Answering Definition Questions Using Web Knowledge Bases

Zhushuo Zhang, Yaqian Zhou, Xuanjing Huang, and Lide Wu

Department of Computer Science and Engineering, Fudan University, Shanghai, China, 200433
{zs_zhang, zhouyaqian, xjhuang, ldwu}@fudan.edu.cn

Abstract. This paper presents a definition question answering approach, which is capable of mining textual definitions from large collections of documents. In order to automatically identify definition sentences from a large collection of documents, we utilize the existing definitions in the Web knowledge bases instead of hand-crafted rules or annotated corpus. Effective methods are adopted to make full use of Web knowledge bases, and they promise high quality response to definition questions. We applied our system in the TREC 2004 definition question-answering task and achieved an encouraging performance with the F-measure score of 0.404, which was ranked second among all the submitted runs.

1 Introduction

When people want to learn an unknown concept from a large collection of documents, the most commonly used tools are the search engines. They submit a query to a search engine system, and the search engine returns a number of pages related to the query terms. Usually, the pages returned are ranked mainly based on keywords matching rather than their relevance to the query terms. The users have to read a lot of returned pages to organize the information they wanted by themselves. This procedure is time-consuming, and the information acquired is not concentrative. The research of Question Answering (QA) intends to resolve this problem by answering user's questions with exact answers.

Questions like "Who is Colin Powell?" or "What is mold?" are definition questions [3]. Their relatively frequent occurrences in logs of Web search engines [2] indicate that they are an important type of question. The Text REtrieval Conference (TREC) provides an entire evaluation for definition question answering from TREC2003. A typical definition QA system extracts definition nuggets that contain the most descriptive information about the question target (the concept for which information is being sought is called the target term, or simply, the target) from multiple documents.

Until recently, definition questions remained a largely unexplored area of question answering. Standard factoid question answering technology, designed to extract single answers, cannot be directly applied to this task. The solution to this interesting research challenge will involve the techniques in related fields such as information extraction, multi-document summarization, and answer fusion.

In order to extract definitional nuggets/sentences, most systems use various pattern matching approaches. Kouylekov *et al.* [10] relied on a set of hand-crafted rules to

R. Dale et al. (Eds.): IJCNLP 2005, LNAI 3651, pp. 498 – 506, 2005.

find definitional sentences. Sasha *et al.* [12] proposed to combine data-driven statistical method and machine learned rules to generate definitions. Cui *et al.* [7] used soft patterns, which were generated by unsupervised learning. Such methods require human labor to construct patterns or to annotate corpus more or less.

Prager *et al.* [8] try to solve this problem through existing technology. They decompose a definition question into a series of factoid questions. The answers to the factoid questions are merged to form the answer to the original question. However, the performance of their system on the TREC definition QA task is unsatisfactory. They need a more proper framework to determine how to generate these follow-up questions [8].

Some systems [1] [7] [9] statistically rank the candidate answers based on the external knowledge. They all adopt a centroid-based ranking method. For each question, they form one centroid (i.e., vector of words and frequencies) of the information in the external knowledge, and then calculate the similarity between the candidate answer and this centroid. The ones that have large similarity are extracted as the answers to this question.

Among the abundant information on the Web, Web knowledge bases (KBs) are one kind of most useful resource to acquire information. Dictionary definitions often supply knowledge that can be exploited directly. The information from them can model the interests of a typical user more reliably than other information. So we go further in identifying and selecting definition sentences from document collection using Web knowledge bases.

Our work differs from the above in that we make use of the Web knowledge bases in a novel and effective way. Instead of using centroid-based ranking, we try to find out more effective methods in ranking the candidate sentences. We consider the relationship and the difference between the definitions from different knowledge sources. In our first algorithm, we calculate the similarity scores between the candidate sentence and the definitions from different knowledge bases respectively, and merge these scores to generate the weight of this candidate sentence. In another algorithm, we first summarize the definitions from different KBs in order to eliminate the redundant information, and then use this summary to rank the candidate sentences. We have applied our approaches to the TREC 2004 definition question-answering task. The results reveal that these procedures can make better use of the knowledge in the Web KBs, and the extracted sentences contain the most descriptive information about the question target.

The remainder of the paper is organized as follows. In Section 2, we describe the system architecture. Then in Section 3 we give the details of our definition extraction methods. The evaluation of our system and the concluding remarks are given in Section 4 and Section 5 respectively.

2 System Architecture

We adopt a general architecture for definition QA. The system consists of five modules: question processing, document processing, Web knowledge acquisition, definition extraction, and an optional module corpus information acquisition. The process of answering a definition question is briefly described as follows.

Firstly, a definition question is input, and the question processing module identifies the question target from this question. The so called target or target term is a term for which information is being sought (e.g., the target of the question "What is Hale Bopp comet?" is "Hale Bopp comet".) The target term is the input for document processing module and knowledge acquisition module.

Secondly, the document processing module generates the candidate sentence set according to this target term. This module has three steps, document retrieval, relevant sentence extraction and redundancy removal. In the first step, the documents that relevant to the target are retrieved from the corpus. In the second step, the sentences that relevant to the target are extracted from these documents. We first cut the documents into sentences, and delete the irrelevant sentences by a few heuristic rules. In the third step, the redundant sentences are deleted by calculating the percentage of shared content words between sentences. After these three steps, we get the candidate sentence set.

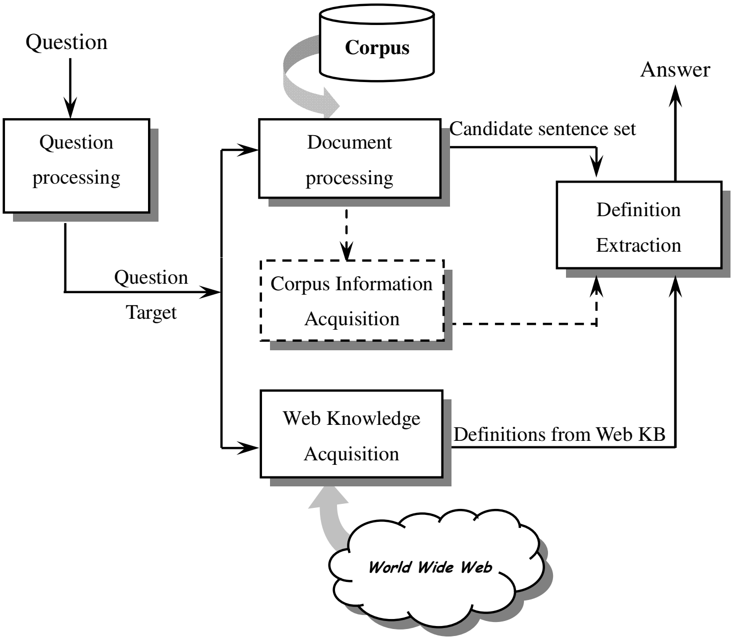

Fig. 1. System architecture

Thirdly, the Web knowledge acquisition module acquires the definitions of the target term from the Web knowledge base. If we can find definitions from these sources (the Web KBs we used will be described in Section 3.1), we use them to rank the candidate sentences set.

At last, the definition extraction module extracts the definition from the candidate sentence set based on the knowledge which is got from the Web knowledge base.

In very few situations, no definitions can be found from the Web KBs, and the module named "corpus information acquisition" is adopted to form the centroid of the candidate sentence set. We rank candidate sentences based on this centroid. The sentences that have high similarity with this centroid are extracted as the answers to the question. The assumption is that words co-occurring frequently with the target in the corpus are more important ones for answering the question.

The system architecture is illustrated in Fig.1.

In this paper, we focus on how to make use of the Web KBs in extracting definition sentences, so we will describe the detail of the definition extraction module below.

3 Definition Extraction Based on Web Knowledge Bases

3.1 Web Knowledge Base

There are lots of specific websites on the Web, such as online biography dictionaries or online cyclopaedias. We can get biography of a person, the profile of an organization or the definition of a generic term from them. We call this kind of website Web knowledge base (KB). The definitions from them often supply knowledge that can be exploited directly. So we answer definition questions by utilizing the existing definitions in the Web knowledge bases. The results of our system reveal that the Web knowledge bases are quite helpful to answering definition questions.

Usually, different knowledge bases may pay attention to different kind of concept, and they may have different kind of entries. For example, the biography dictionary (www.s9.com) is a dictionary that covers widely on biography of people, and other KBs may pay attention to other kinds of concept. We choose several authoritative KBs that cover different kinds of concept to achieve our goal.

The Web knowledge bases we used are the Encyclopedia(www.encyclopedia.com), the Wikipedia(www.wikipedia.com), the Merriam-Webster dictionary (www.mw.com), the WordNetglossaries (www.cogsci.princeton.edu/cgi-bin/webwn) and a biographies dictionary (www.s9.com).

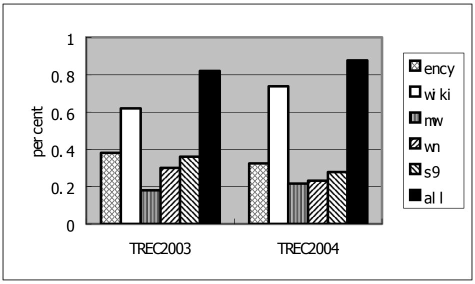

Fig. 2. The Web KBs' coverage of TREC data

These Web KBs can cover most of the target terms, and the definitions in them are exact and concise.This can be confirmed from the experiment on TREC's data set. Fig.2 gives our experiment results on the TREC 2003 and TREC 2004's definition question sets, which have 50 and 65 target terms respectively. The "ency", "wiki", "mw", "wn" and "s9" stand for the five online KBs we have used. Each column represents the percent of the target terms that can be found in the corresponding online knowledge base. The column marked "all" represents the percent of the target terms that can be found in at least one of these five online knowledge bases.

It is easy to see that a high coverage can be got by using these Web knowledge bases. In the Section 3.2 and Section 3.3 we will show how to use these KBs, and in Section 4 we can see that it boosts the performance of the system significantly.

3.2 Definition Extraction Based on *GDS*

As mentioned above, we may get most of the submitted target terms' definitions by utilizing multiple Web KBs. One target may find its definitions in more than one knowledge base. Are all of them useful? The experimental data tells that, the different definitions belonging to one target differ from each other in some degree. They are short or long, concise or detailed.

Considering the above factor, we try to utilize all of the definitions from different Web KBs to accomplish our task. For one target term, the definitions from all Web knowledge bases compose its "general definition set", which is abbreviated to ***GDS***. Each element of this set is a definition from one Web knowledge base, so the number of the elements in this set is the same as the number of the Web KBs we used. When we cannot find its entry in a certain Web KB, its corresponding element will be an empty string.

For each target, its candidate sentence set is expressed as $S_A = \{A_1, A_2, \ldots, A_m\}$, where A_k (k=1..m) is a candidate sentence in the set and *m* is the total number of the candidate sentences.

GDS is expressed as $S_{GD} = \{D_1, D_2, \ldots, D_n\}$, where D_k (k=1..n) is the definition of the target from the *k*th knowledge base, and *n* is the number of the knowledge bases. D_k may be an empty string when the target has no definition in the knowledge base *k*. In this algorithm, we rank the candidate sentences set $S_A = \{A_1, A_2, \ldots, A_m\}$ using S_{GD}.

Let S_{ij} be the similarity of A_i and D_j. The similarity is the tf.idf score, where the candidate sentence A_i and the definition D_j are all treated as a bag of words. The tf.idf function we used is described in [5].

For each candidate sentence A_i in the set S_A, we calculate its score based on the ***GDS*** as follows:

$$score_i = \sum_{j=1}^{n} w_j S_{ij} \quad (\sum_{j=1}^{n} w_j = 1) . \tag{1}$$

The weights w_j are fixed based on experiment, considering the authoritativeness of the knowledge base from which D_j comes. The sentences of set S_A are ranked based on this score, and the top ones are chosen as the definition of the target term.

3.3 Definition Extraction Based on *EDS*

As we have seen, for a target term, different definitions in its "general definition set" may overlap in some degree. We intent to modify this set by merging its elements into one concise definition. We extract the essential information from the "general definition set" to form the "essential definition set", which is abbreviated to ***EDS***.

EDS is expressed as $S_{ED} = \{d_1, d_2, \ldots, d_l\}$, where each element d_k (k=1..*l*) is an essential definition sentence about the target, and *l* is the number of the essential definition sentences. We hope that each element can tell one important aspect of the target term, and the whole "essential definition set" may contain as much information as ***GDS*** but no redundant information.

We try to use an automatic text summary technique [11] to get ***EDS***. This technique is based on sentence's weight and similarity between sentences. Firstly, calculate the weights of all sentences and similarities between any two sentences, and then extract sentence based on these weights. After one sentence has been extracted, calculate the new weights of the remained sentences based on their similarities. Iterate the above procedure until the extracted sentences reach the required length. More detail of this technique can be found in [11]. In this section we will try to use the "essential definition set" to extract definitions from the candidate sentence set.

1. Initially set the result set A={ }, and *i*=1.

2. For the element d_i in the set S_{ED}:

First get the similarity between d_i and A_j (j=1..m), which is expressed as S_{ij}.

Then let $S_{ik} = \max\{S_{i1}, S_{i2}, \ldots, S_{im}\}$. If $S_{ik} > minsim$, then add A_k to the set A and delete A_k from the set S_A .

3. If $\sum_{k=1}^{m'} L(A_k) > \max\_length$ or *i* equals to *l*, the algorithm ends; otherwise, *i* = *i* +1, go to step2.

Fig. 3. Definition extraction using *EDS*

The algorithm was showed in Fig.3. The candidate sentence set is also expressed as $S_A = \{A_1, A_2, \ldots, A_m\}$, where A_k (k=1..m) is a candidate sentence in the set and *m* is the total number of the candidate sentences. The similarity S_{ij} is calculated as the same as in Section 3.2. $L(A_k)$ represents the length of string A_k in character and m' is the number of elements in set A. The parameters *max_length* and *minsim* were empirically set based on TREC's definition question set. The last result is set A, where $A=\{A_1, A_2, \ldots, A_{m'}\}$.

4 Evaluation

In order to get comparable evaluation, we apply our approach to TREC2004 definition QA task. We can see that our approach is an effective one compared with peer systems in this competitive evaluation.

In this section we present the evaluation criterion and system performance on TREC task, and discuss the effectiveness of our approach.

4.1 Evaluation Criterion

The TREC evaluation criterion [3] is summarized here for the purpose of discussing the evaluation results.

For an individual definition question, there is a list of essential nuggets and acceptable nuggets provided by TREC. These given nuggets are used to score the definition generated by the system.

An individual definition question will be scored using nugget recall (R) and an approximation to nugget precision (P) based on length. In particular,

R = # essential nuggets returned in response/# essential nuggets

P is defined as: if length < allowance, P = 1
else P=1-[(length-allowance)/length]

where allowance = 100*(# essential+acceptable nuggets returned)
length = total # non-white-space characters in answer strings

The F measure is:

$$\mathrm{F} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \quad . \tag{2}$$

where β value is fixed three in TREC 2004, and we also use three to get comparable result.

The score of a system is the arithmetic mean of F-measure scores of all the definition questions output by the system.

4.2 Effectiveness of Web Knowledge Bases

To compare the effectiveness of the Web knowledge bases, we experimented on the TREC 2004 definition question set. The result can be seen in Table1.

Table 1 shows the F-measure scores of our two algorithms and the baseline method. It also shows the median of the scores of all participating systems in TREC 2004. The baseline method is: for an input question, form the candidate sentence set by using the approach described in Section 2. Then put the sentence of this set into the answer set one by one until all the sentences in the candidate sentence set are considered or the answer length is greater than a pre-fixed length (we set the length 3000 characters in our experiment).

We can see that our two algorithms all outperform the median and the baseline method which does not use Web knowledge bases. In conclusion, the Web knowledge bases are effective resources to definition question answering.

Table 1. The F- measure score of the baseline method, the median system in TREC2004, and our two methods on TREC 2004 data set

	Baseline method	Median	Ranking using ***GDS***	Ranking using ***EDS***
F-measure score (β=3)	0.231	0.184	0.404	0.367

4.3 Definition Extraction Based on *GDS* vs. Based on *EDS*

As we have mentioned, we have tried two algorithms in the definition extraction module, which are based on ***GDS*** and ***EDS*** respectively. The performance of these algorithms is shown in Table 2.

Table 2. Performance of our three runs on the three types of quesitions and on the whole 64 questions of TREC 2004

	Num Q	Run A	Run B	Run C
all	64	0.404	0.389	0.367
PERSON	23	0.366	0.372	**0.404**
ORG	25	**0.413**	0.389	0.326
THING	16	**0.446**	0.415	0.379

We have submitted three runs in TREC2004, which were generated by using different algorithm in the definition extraction module. Run A and run B were generated by using ***GDS*** with slightly different weights in formula (1), and run C was generated by using ***EDS***. All the 64 questions are divided into three classes based on the entity types of the targets, which are person, organization and other thing. Table 2 shows the three runs' F-measure scores on these three types and their overall score on the whole 64 questions.

Two algorithms' F-measure scores are all among the best of total 63 runs. Run C's score on the "PERSON", 0.404 is the highest of our three runs on this type. Run A does better on the types named "ORG" and "THING". We can say that these two algorithms contribute to different kinds of target terms. Dividing definition questions into different subclass and processing them with different methods could be a proper direction.

Considering the score on all the 64 questions, the former algorithm is slightly higher than the latter one. However, the result of the latter one is also encouraging. Since the "essential definition set" contain the important information and less redundancy, it has the potential to get the answers, which are not only concise but also have wide coverage about the target. We believe it is an appropriate way to extract the high quality definitions. A preliminary analysis shows that the major problem is how to improve the quality and the coverage of the essential definition set. We believe that the performance could be boosted through improving this technique.

In conclusion, we can say that our methods can make better use of the external knowledge in answering definition question.

5 Conclusions

This paper proposes a definition QA approach, which makes use of Web knowledge bases and several complementary technology components. The experiments reveal that the Web knowledge bases are effective resources to definition question answering, and the presented method gives an appropriate framework for answering this kind of question. Our approach has achieved an encouraging performance with the F-measure score of 0.404, which is ranked second among all the submitted runs in TREC2004.

Since definitional patterns can not only filter out those statistically highly-ranked sentences that are not definitional, but also bring those definition sentences that are written in certain styles for definitions but are not statistically significant into the answer set. [6] In the future work, we will employ some pattern matching methods to reinforce our existing method.

Acknowledgements

This research was partly supported by NSF (Beijing, China) under contract of 60435020, and Key Project of Science and Technology Committee of Shanghai under contract of 035115028.

References

1. Abdessamad Echihabi, Ulf Hermjakob, Eduard Hovy: Multiple-Engine Question Answering in TextMap. In Proceedings of the Twelfth Text REtreival Conference. NIST, Gathersburg, MD (2003) 772–781
2. Ellen M. Voorhees: Overview of the TREC 2001 question answering track. In Proceedings of the Tenth Text REtreival Conference. NIST, Gathersburg, MD (2001) 42–51
3. Ellen M. Voorhees: Overview of the TREC 2003 Question Answering Track. In Proceedings of the Twelfth Text REtreival Conference. NIST, Gathersburg, MD (2003) 54–68
4. Ellen M. Voorhees: Evaluating answers to definition questions. In Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (2003) Volume 2: 109–111
5. G.Salton, C. Buckley: Term weighting approaches in automatic text retrieval. Information Processing and Management (1988) 24(5): 513–523
6. 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)
7. Hang Cui, Keya Li, Renxu Sun, Tat-Seng Chua, Min-Yen kan: National University of Singapore the TREC-13 Question Answering Main Task. In Proceedings of the Thirteenth Text REtreival Conference. NIST, Gathersburg, MD (2004)
8. J. M. Prager, Jennifer Chu-Carroll, Krzysztof Czuba, Christopher Welty, Abraham Ittycheiach, Ruchi Mahindru: IBM's PIQUANT in TREC2003. In Proceedings of the Twelfth Text REtreival Conference. NIST, Gathersburg, MD (2003) 283–292
9. Jinxi Xu, Ana Licuanan, Ralph Weischedel: TREC2003 QA at BBN: Answering definitional Questions. In Proceedings of the Twelfth Text REtreival Conference. NIST, Gathersburg, MD (2003) 98~106
10. Milen Kouylekov, Bernardo Magnini, Matteo Negri, Hristo Tanev: ITC-irst at TREC-2003: the DIOGENE QA system. In Proceedings of the Twelfth Text REtreival Conference. NIST, Gathersburg, MD (2003) 349–357
11. Qi Zhang, Xuanjing Huang, Lide Wu: A New Method for Calculating Similarity between Sentences and Application on Automatic Text Summarization. In Proceedings of the first National Conference on Information Retrieval and Content Security (2004)
12. Sasha Blair-Goldensohn, Kathleen R. McKeown, Andrew Hazen Schlaikjer: A hybrid approach for QA track definitional questions. In Proceedings of the Twelfth Text REtreival Conference. NIST, Gathersburg, MD (2003) 185–192