LETTER

Descriptive Question Answering with Answer Type Independent Features

Yeo-Chan YOON[†], **Chang-Ki LEE**[††], **Hyun-Ki KIM**[†], **Myung-Gil JANG**[†], **Pum Mo RYU**[†], ***Nonmembers***, ***and*** **So-Young PARK**[†††a)], ***Member***

SUMMARY In this paper, we present a supervised learning method to seek out answers to the most frequently asked descriptive questions: reason, method, and definition questions. Most of the previous systems for question answering focus on factoids, lists or definitional questions. However, descriptive questions such as reason questions and method questions are also frequently asked by users. We propose a system for these types of questions. The system conducts an answer search as follows. First, we analyze the user's question and extract search keywords and the expected answer type. Second, information retrieval results are obtained from an existing search engine such as Yahoo or Google. Finally, we rank the results to find snippets containing answers to the questions based on a ranking SVM algorithm. We also propose features to identify snippets containing answers for descriptive questions. The features are adaptable and thus are not dependent on answer type. Experimental results show that the proposed method and features are clearly effective for the task.

key words: *descriptive question answering*

1. Introduction

Question answering is a task that gives answers for natural language question such as "What is X?" or "Who's the president of South Korea?"

Since the introduction of the TREC QA track, many related works for question answering have been carried out. TREC encouraged the study of question answering for many types of questions. Depending on the length of the answer, there can be roughly two types of questions in TREC QA tracks, factoid questions, which find short answers, and definition questions, which are some of the most frequently asked descriptive questions. However, other frequently asked descriptive questions such as reason questions and method questions are rarely focused on in TREC tasks.

TREC formalized definitional question answering as extracting and combining answers from multiple documents [1] for questions such as "What are fractals?" or "Who is Andrew Carnegie?" A variety of approaches have been proposed for definitional question answering tasks [2], [3].

Manuscript received December 9, 2011.
Manuscript revised March 16, 2012.

[†]The authors are with Speech/Language Information Research Center, ETRI, Gajeong-dong, Yuseong-gu, Daejon, Korea.
[††]The author is with the Department of Computer Science, Kangwon National University, 192–1 Hyoja2-dong Chuncheon-si Gangwon-do, Korea.
[†††]The author is with the Division of Digital Media Technology, SangMyung University, 7 Hongji-dong, Jongro-gu, Seoul, Korea.
a) E-mail: ssoya@smu.ac.kr (Corresponding author)

DOI: 10.1587/transinf.E95.D.2009

While these works focused on extracting descriptive phrases for a definition term from multiple documents and combining them, other works simply ranked descriptive phrase candidates or snippets [4], [5]. Miliaraki et al. [4] trained an SVM model and ranked snippets from IR engine results based on SVM scores to find answers for definition questions. They picked an n-gram of tokens that occur immediately before or after the definition term and exploited them as features. Xu et al. [5] employed ranking SVM as an ordinal regression model and ranked definition candidates extracted with heuristics rather than combining them. They also insisted that definitions usually extracted from different documents describe target terms from different perspectives, and thus it is not easy to combine them together. In addition to the n-gram feature used by Miliarkai et al. [4] they used various features such as number of sentences in a paragraph.

Definition question answering has been studied continuously. On the other hand, there are few studies on question answering for other important descriptive questions such as reason questions and method questions. Oh et al. [6] proposed a method for descriptive question answering in an encyclopedia. They classified expected answer types for descriptive questions into ten types, such as definition, method, reason, function, and kind. For each question type, they manually built patterns and extracted answers from the encyclopedia.

In this paper, we propose a question answering system for most frequently asked descriptive questions: definition, reason, and method questions. We exploit an existing Web search API to find snippets related to a question. We then rank the snippets that include answers. There are some advantages in providing snippets as answers rather than extracting and combining sentences or phrases. As Xu et al. [5] stated, different documents have different perspectives; therefore, contextual information is required for a comprehensive understanding, and users can glimpse the context from the searched snippets. Moreover, users can read a whole document using a provided source URL and check other related information.

We employ the ranking SVM model to rank search results and develop useful features. The features are not dependent on expected answer types of descriptive questions, and so can be applied to all ranking models for descriptive questions attempted in this paper. Therefore, various types of descriptive question could be covered by the proposed approach with less effort.

Our experimental results indicate that our approach and proposed features are very effective for finding snippets that include answers for descriptive questions.

2. Finding Descriptive Snippets

Our method consists of three major components: a question analysis component, Web search component, and descriptive snippet ranking component. In the question analysis component, the expected answer type is recognized for the question. In the Web search component, the question is converted into a query to search related documents. Searched results are ranked in the descriptive snippet ranking component based on the ranked SVM confidence scores.

2.1 Question Analysis and Web Search

The important role of a question analysis is identifying the expected answer type. We used a previous work [6] to recognize the answer type. The work in [6] used 724 lexico-syntactic patterns such as "What's the reason X" and "How to X" to recognize the answer type. The previous work shows 87.03% f-score. However, we appended 205 patterns to the previous work to make its f-score 100% in order to evaluate the exact performance of our method. The recognized answer type is used for choosing the ranking SVM models. We built three models for each question types: reason, definition, and method questions.

We exploited the Yahoo search API for IR. We converted a natural language question into a search query. We searched twice with two queries to find more related snippets for the question. The first query consists of words occurring in the question other than stop words such as 'what'. We also appended clue words such as 'reason', 'definition', and 'methods' that occur most frequently in descriptive questions to the first query, and used it as the second query. Figure 1 illustrates the results of the question analysis module.

2.2 Ranking Search Results

Most Web search engines give results in the form of a snippet, which is a summary of documents. We converted a snippet into a feature vector and ranked it using a supervised learning method. Joachims [7] proposed a ranking SVM model and used it for ranking the functionality of a search engine. We employed the ranking SVM and ranked the snippets based on its score output. Given an instance of a query and snippet x pair, the equation for the ranking SVM is as follows [5]:

$$U(x) = w^t x, \tag{1}$$

where w represents a vector of weights. A higher $U(x)$ score indicates that it is more possible that a snippet includes an answer for a question.

The construction of a ranking SVM needs labeled training data. We collected 300 questions (100 questions for each question type) and built 51,701 question-snippet pairs for a training set. A total of 4,700 question-snippet pairs, including answers for a question, are used as positive samples, and other pairs are used as negative samples. The language of the set is Korean. We also built a set of 27,458 question-snippet pairs for 150 questions (50 questions for each question type) for the test set. Table 1 shows the details of the number of each set.

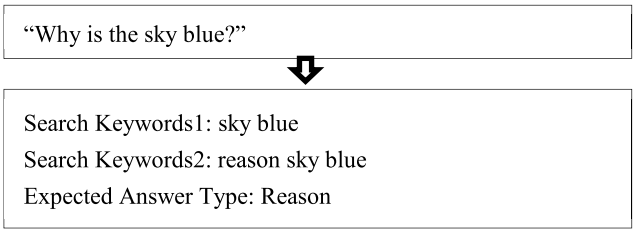

Fig. 1 Results of question analysis.

Table 1 Number of samples for each set.

	Positive (Training)	Negative (Training)	Positive (Test)	Negative (Test)
Definition	2,087	15,112	1,253	9,029
Reason	1,384	21,655	703	10,817
Method	1,229	10,234	776	4,880
Total	4,700	47,001	2,732	24,726

We tried to discover features that are independent and adaptable to every descriptive answer type. First, we used n-grams ($n \in \{1, 2, 3\}$) of tokens that occur immediately before or after the target term as a feature, which is used in [4]. This feature is useful for detecting patterns as follows:

ap (answer phrase) known as *tn* (target term)
e.g., "Motorola Droid3 known as Solana"

Detecting target terms of a question is a very difficult issue. Many of the previous works for definition questions assumed that the target term is pre-identified or pre-acquired as an input [4], [5]. However, we simply treated all nouns from a question as target words except stop words.

The n-gram feature misses some patterns that occur far from the target terms. In the example below, the pattern 'as it mean' has a twelve-word distance with the target word, 'Alesio'.

Q: What does Alesio mean?
A: Alesio is a perfect name for a brave pet or a guard dog as it means "defender".

Therefore, we also considered n-gram which occurs far from the target terms. However, some patterns such as "*tn* is the *ap*"[†] occur right after target terms and deteriorate the performance if the n-gram is used without consideration of the distance from target terms. Accordingly, we also considered the distance from the target terms. For example, we

[†]For instance, "*CPU* is the *portion of a computer system*".

can acquire n-gram with the distance for the above snippet as follows:

Unigram: (is, 0), (a, 1), . . . , *(as, 12), (it, 13), (means, 14)*, . . .
Bigram: (is a, 0), (a perfect, 1) . . . , (as it, 12), . . .
Trigram: (is a perfect, 0), . . . , (as it means, 12), . . .

However, there could be a huge number of possible patterns as the distance can be any number. Therefore, we normalized the distance value lager than 5 as "FARNEXT" and the distance value less than −5 as "FARPREV" because of data sparseness problem.

We also exploited the ranking of Web search engine results as a feature. Existing Web search engines are well tuned and provide related documents for a query. Thus, they can be used as a useful measure of relevance. We normalized the ranking as a value between 0 and 1 and exploited it as a feature.

3. Experimental Results

In the experiment, top N precision (N = 1 or 3 or 5) is used as a measure of evaluation.

$$\textit{Top N Precision} = \frac{|\textit{questions whose top N ranked snippets contains answer}|}{|\textit{all questions in data set}|}$$

As one baseline method, we used the results of a Web search engine. It could be considered that the last step of the proposed approach, the descriptive snippet ranking, be skipped. As other baseline methods, we exploited features proposed in [5] and [4]. Miliaraki et al. used an n-gram as a main feature [4]. While they used words occurring both before and after the target word as an n-gram, Xu et al. [5] only considered an n-gram occurring after the target word. They also used various features in addition to an n-gram such as "number of words in a paragraph," "does target word occur at the beginning of the paragraph," "target word reoccurs in the paragraph," "number of sentences in the paragraph," "number of words in the paragraph," and "number of the adjectives in the paragraph." Xu et al.'s approach used certain features depending on the English language, such as "target word contains 'of' "; however, we used a Korean set, and therefore ignored such features.

The results reported in Table 2 indicate that the proposed features outperform baseline features. N-grams after the target term showed better performance than n-grams before and after the target term; because the answer phrases tend to appear after the target term rather than before the target term. Given the sentence "CPU is the portion of a computer system", for example, the answer phrase appears after the target term '*CPU*'. However, the performance is significantly increased with the distance information as long distance word patterns could be considered with the distance feature.

Xu et al. [5] proposed features for definition questions. However, these features also work for other descriptive questions because they represent how much the paragraph is well expressed, or how much the paragraph is related to the target term. Therefore, these features generally improved the performance for descriptive questions. The performance of the proposed system is also elevated with these features.

Ranking of a Web search engine feature elevated the performance when added to an n-gram and distance features. These results show that considering the relevance is also effective for finding descriptive answers.

Table 3 shows the TOP1 performance for each answer type. The performance for a definition question was higher than the other types. For a definition question, there were more answer snippets among Web search results than the other types. Table 4 shows the details. Moreover, the average number of terms in a definition question is lower than other types. We assumed all terms in question as target

Table 2 System performances with various features.

			TOP1	TOP3	TOP5
Baseline	(1)	Web results	0.267	0.500	0.640
Miliaraki et al.[4]	(2)	n-grams before and after target term	0.327	0.513	0.593
Xu et al.[5]	(3)	n-grams after target term	0.353	0.580	0.647
	(4)	(3) + heuristic features	0.380	0.580	0.660
Proposed Model	(5)	(3) + distance	**0.420**	**0.640**	**0.780**
	(6)	(5) + rank	**0.460**	**0.680**	**0.793**
	(7)	(6) + heuristic features	0.467	0.713	0.813

Table 3 Top 1 performances for each answer type.

			Method	Definition	Reason
Baseline	(1)	Web results	0.22	0.30	0.28
Miliaraki et al.[4]	(2)	n-grams before and after target term	0.26	0.44	0.28
Xu et al.[5]	(3)	n-grams after target term	0.28	0.38	0.40
	(4)	(3) + heuristic features	0.28	0.42	0.44
Proposed Model	(5)	(3) + distance	**0.34**	**0.46**	**0.46**
	(6)	(5) + rank	**0.42**	**0.50**	**0.46**
	(7)	(6) + heuristic features	0.42	0.48	0.50

Table 4 Statics on each question type.

	Method	Definition	Reason	All
average # of answers	15.52	25.06	14.06	18.21
average # of terms in questions	3.14	1.42	2.82	2.46
average # of IR results	113.62	201.41	229.78	179.38

Table 5 Sign test results (p-value).

	TOP1	TOP3	TOP5
(1) vs (6)	9.94E-05	3E-04	7E-04
(2) vs (6)	0.00259	7.27606E-05	2.60897E-06
(3) vs (6)	0.01602	0.01582	0.00038
(4) vs (6)	5.03E-06	3.04E-08	1.9674E-08
(5) vs (6)	0.02876	0.0167	0.264443

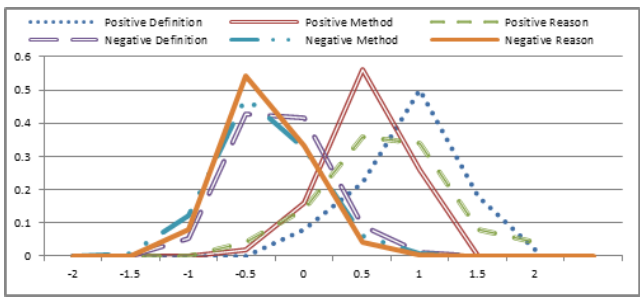

Fig. 2 Distribution of SVM scores for data set.

terms. If a question has only one term, the term should be the target term while some terms may be miss-assumed as the target term if the question has a few terms. Therefore, more terms in the question would deteriorate the system performance. For these reasons, the performance for definition questions is better than the other types of questions. However, the proposed features are still generally effective for various question types.

We also conducted a sign test on features proposed over the baseline features (cf., Table 5). We performed a paired-t test for significance. From this, we found that our proposed model significantly outperforms the models using Web results and an n-gram with Xu's features ($p < 0.01$), and the n-gram model ($p < 0.05$). The model using the ranking feature also significantly outperformed the model without the feature in Top1 and Top3 ($p < 0.05$) measures.

Figure 2 shows the distribution of positive and negative samples for each question type (vertical axis) over the SVM score (horizontal axis). There is a huge difference between the distributions for positive and negative samples. In general, the scores for the positive samples are higher than the negative samples. From this, we found that finding answers based on the scores of ranking SVM is quite reasonable.

4. Conclusion

In this paper, we proposed a new question answering method for the most frequently asked descriptive questions. From Web search results, the proposed approach uses a ranking SVM to rank snippets to find answers for a question. Answer type independent features are discovered for the method. With these features, we can easily apply our approach to various types of descriptive questions. Experimental results indicate that our proposed method performs significantly better than the baseline methods. The results also show that the proposed method is effective on all types of descriptive questions tested. As future works, we will apply and expand our approach to other descriptive questions such as origin and function. We also will try to discover more features for descriptive questions.

References

[1] E. Voorhees, "Overview of the TREC 2003 question answering track," Proc. 12th Annual Text Retrieval Conference, 2003.
[2] S. Blair-Goldensohn, K.R. McKeown, and A. HazenSchlaikjer, "A hybrid approach for QA track definitional questions," Proc. TREC 2003, pp.336–343, 2003.
[3] J. Xu, A. Licuanan, and R. Weischedel, "TREC2003 QA at BBN: Answering definitional questions," Proc. Twelfth Text Retrieval Conference (TREC 2003).
[4] S. Miliaraki and I. Androutsopoulos, "Learning to identify single-snippet answers to definition questions," 20th International Conference on Computational Linguistics (COLING 2004), pp.1360–1366, 2004.
[5] J. Xu, Y.B. Cao, H. Li, and M. Zhao, "Ranking definitions with supervised learning methods," Special Interest Tracks and Posters of the 14th International Conference on World Wide Web, Chiba, Japan, May 2005.
[6] H.J. Oh, C.H. Lee, H.J. Kim, and M.G. Jang, "Descriptive question answering in encyclopedia," Proc. ACL Interactive Poster and Demonstration Sessions, pp.21–24, Ann Arbor.
[7] T. Joachims, "Optimizing search engines using clickthrough data," Proc. 8th ACM Conference on Knowledge Discovery and Data Mining, 2002.
[8] C.K. Lee and M.G. Jang, "Fast training of structured SVM using fixed-threshold sequential minimal optimization," ETRI J., vol.31, no.2, pp.121–128, April 2009.