A Novel Pattern Learning Method for Open Domain Question Answering

Yongping Du, Xuanjing Huang, Xin Li, and Lide Wu

Department of Computer Science, Fudan University
Shanghai, 200433
{ypdu,xjhuang,lixin,ldwu}@fudan.edu.cn

Abstract. Open Domain Question Answering (QA) represents an advanced application of natural language processing. We develop a novel pattern based method for implementing answer extraction in QA. For each type of question, the corresponding answer patterns can be learned from the Web automatically. Given a new question, these answer patterns can be applied to find the answer. Although many other QA systems have used pattern based method, however, it is noteworthy that our method has been implemented automatically and it can handle the problem other system failed, and satisfactory results have been achieved. Finally, we give a performance analysis of this approach using the TREC-11 question set.

1 Introduction

Question answering has recently received much attention from the natural language processing communities. The Text Retrieval Conference (TREC) Question Answering track provides a large-scale evaluation for open domain question answering systems. The goal of question answering is to retrieve answers to questions rather than documents as most information retrieval systems currently do.

An integrated QA system has three main components as shown in Fig.1. The first is question analysis that determines the answer type and translates natural language questions into queries for the search engine. The second is search module that retrieves relevant documents or snippets from the document collection, which can potentially answer the question. The third component, answer extraction, analyzes these documents or snippets and extracts answers from them. For example, question "What is the largest city in Germany?" is the input of the QA system and the answer "Berlin" is returned as the output.

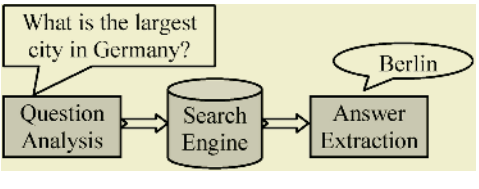

Fig. 1. Architecture of QA System

K.-Y. Su et al. (Eds.): IJCNLP 2004, LNAI 3248, pp. 81–89, 2005.

We develop a novel pattern based method for implementing answer extraction. For each type of question, the corresponding answer patterns can be learned from the Web automatically. Given a new question, these answer patterns can be applied to find the answer.

Many other question answering systems have used pattern based method. ISI [1] and Singapore-MIT Alliance[2] have implemented pattern learning of different question type for QA. For instance, the pattern "in <ANSWER>'s <NAME>" is learned for the question type "LOCATION", where "<NAME>" denotes the question term. A serious limitation of these patterns is that it can handle only one question term in the candidate answer sentence and it can't work for more complex questions that require multiple question terms in the answer sentence. InsightSoft[3] has achieved good performance in TREC but their patterns can't be learned automatically. It is noteworthy that our method has been implemented automatically and it can answer questions that require more than one question term in the candidate answer sentence.

The wealth of information on the Web makes it an attractive resource and many systems have make use of the Web knowledge[4][5][6]. We also take advantage of the variety on Internet for learning different answer patterns which are used for answer extraction in QA. Each answer pattern is consisted of the following three parts:

<Q_Tag>+[ConstString]+<A>

Here, <Q_Tag> stands for the key phrases of question and we will introduce them later. <A> stands for the answer, and any string holding the position will be extracted as the answer. [ConstString] is a sequence of words.

This paper first introduces the question analysis for <Q_Tag> identification in section 2, and then presents the process of learning answer patterns in section 3, following answer extraction with these answer patterns in section 4, finally, we give the performance analysis in section 5.

2 Question Analysis

We define a set of symbols to represent question as illustrated in Table 1, which are the objects or events the question asks about.

The symbol set of Q_Tag includes four kinds of symbol: Q_Focus, Q_NameEntity, Q_Verb and Q_BNP. Here, Q_NameEntity includes different name entity symbols, such as Q_LCN, Q_PRN and so on. It should be pointed that the noun phrases denoted by the symbol Q_BNP don't include the noun phrases which had been denoted by the symbol Q_Focus and Q_NameEntity.

Q_Focus denotes the key words of the question and it contains the following instances:

- the head word of the noun phrase, which is binding with interrogative
  eg：Which *river* runs through Dublin ?
- the "Noun Phrase" of the question whose sentence structure is "Interrogative + be verb+ Noun Phrase+ …"
  eg：What is *the most populous city* in the United States?
- the "ADJ" of the question whose sentence structure is "How+ ADJ+ be Verb(auxiliary verb)+…"
  eg：How *tall* is Mt. Everest ?

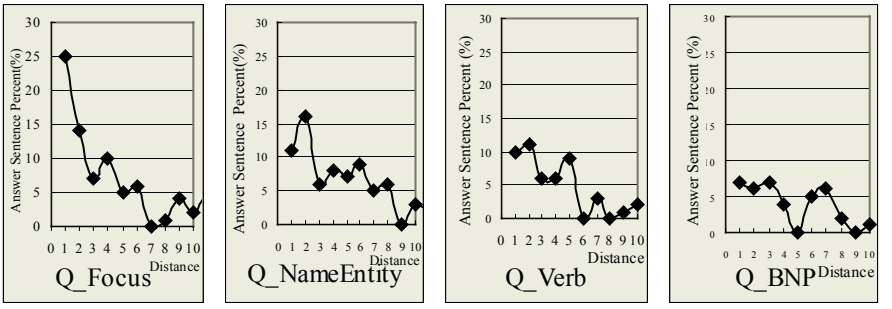

Fig. 2. Distribution of Q_Tag Around Answer

Q_NameEntity, Q_Verb and Q_BNP are analyzed from our Name Entity tagger, Parser and the BNP Chunking tool respectively.

Of all the TREC questions(TREC8-TREC11), we select the answer sentences of 182 questions and these questions contain all the Q_Tag symbols. The distribution of different Q_Tag symbols around the answer in the sentence is shown in Fig.2, and the distance denotes the word count between Q_Tag and the answer. All these Q_Tag symbols are assigned different weights as shown in Table 1, taking into account the possibilities they appear around the answer.

Table 1. Symbol Set of Question

Q_Tag	Description	Example	Weight
Q_Focus	the key word or phrase representing the object that the question asks about	What country is the holy city of Mecca located in?	4
Q_NameEntity (Q_LCN Q_PRN…)	the name entity of the question	What country is the holy city of Mecca located in?	3
Q_Verb (Q_BeVerb Q_DoVerb)	the main verb of the question	What country is the holy city of Mecca located in?	2
Q_BNP	the noun phrase of the question	What country is the holy city of Mecca located in?	1

In our system we adopt a six-class answer type classification, illustrated in Table 2. Currently our system can achieve the precision of 90%, taking 1893 questions of TREC as test data.

Table 2. Answer Type

LCN (Location)	PRN (Person Name)	ORG (Organization)
NUM (Number)	DAT (Date)	BNP (Noun Phrase)

The question pattern (Q_Pattern) is generated from its Q_Tag symbol set, in other words, every element of the question is replaced with its corresponding Q_Tag, and

Table 3. Sample Question Type

Question Type	Question
[LCN] What Q_BeVerb Q_Focus in Q_LCN ?	What is the largest city in Germany? What is the most populous city in the United States?
[DAT] When did Q_LCN Q_DoVerb Q_BNP ?	When did Hawaii become a state ? When did North Carolina enter the union ?
[NUM] What Q_BeVerb Q_Focus of Q_BNP ?	What is the diameter of a golf ball ? What is the melting point of copper ?
[PRN] Who Q_BeVerb Q_Focus of Q_LCN ?	Who is the prime minister of Australia? Who was the first coach of the Cleveland Browns?

then the classification of questions will be built based on the Q_Pattern and the answer type. Sample question types along with corresponding questions are shown in table 3.

Answer patterns can be learned automatically using the <Q_Tag, Answer> pairs as training examples and then used for answer extraction. For instance, answer pattern *", <A> Q_BeVerb Q_Focus in Q_LCN "* can be used to answer the question "What is the largest city in Germany?" where "Q_Focus" denotes the question term *"the largest city*" and "Q_LCN" denotes the question term *"Germany"*. The answer "Berlin" can be extracted from the snippet "*… , Berlin is the largest city in Germany* and is developing into a metropolis of sciences, arts, …".

3 Pattern Learning and Evaluation

We will explain our approach with the sample below.

Sample question type: [LCN] What Q_BeVerb Q_Focus in Q_LCN ?

Sample question: What is the largest city in Germany ?

Where Q_BeVerb=*"is"*, Q_Focus=*"the largest city"*,Q_LCN=*"Germany"*, and Answer=*"Berlin"*.

3.1 Pattern Learning

The answer patterns of each question type are learned by the following algorithm:

1. Constructing Query: "Q_Tag +Answer" is constructed as the query where Q_Tag includes all kinds of Q_Tag except Q_BeVerb. For example, the query of above sample question is: *"the largest city"+"Germany"+ "Berlin"*.
2. Searching: The query is submitted to the search engine Google, and then the top 100 documents are downloaded.
3. Snippet Selection: The snippets are extracted from the documents for pattern learning, containing 10 words around the answer.
4. Answer Pattern Extraction: Replace the question term in each snippet by the corresponding Q_Tag, and the answer term by the tag <A>. The shortest string containing the Q_Tag and the tag <A> is extracted as the answer pattern. For example, considering the string "…With its 3.4 million inhabitants*, Berlin is the largest*

city in Germany and is developing into a metropolis of sciences, arts, …", the answer pattern "*, <A> Q_BeVerb Q_Focus in Q_LCN* " is extracted.

5. Computing the Weight of Each Answer Pattern: It is computed by the following formula considering the weight of the Q_Tag and the distance between different Q_Tag with the answer. (α=1, β=0.6)

$$Weight_P = \alpha \bullet \frac{1}{Distance} + \beta \bullet \sum_{j=1}^{n} \frac{Weight_{Q\_Tag_j}}{Weight_{Sum}} \tag{1}$$

Here,

$$Distance = \frac{\sqrt{d_1^2 + d_2^2 + \cdots + d_n^2}}{n} \tag{2}$$

$$Weight_{Sum} = \sum_{k=1}^{m} Weight_{Q\_Tag_k} \tag{3}$$

Where, *m* is the number of Q_Tag contained in the question type, *n* is the number of the Q_Tag contained in the answer pattern, d_i is the distance between different Q_Tag and the answer, measured by the count of the distinct words. We discard the patterns whose $Weight_P$ is less than a threshold T (T=0.3).

For each question type, it usually have many questions just as shown in Table 3 and we learn answer patterns for all of them. For the sample question above, we obtain following answer patterns:

, <A> Q_BeVerb Q_Focus in
Q_LCN Q_Focus in Q_LCN , <A>
<A> Q_BeVerb Q_Focus
…

3.2 Pattern Evaluation

Among all these answer patterns we have learned, some of them may extract the wrong answer. For the above sample question, answer pattern "*<A> Q_BeVerb Q_Focus*" can extract candidate answer "*Portland*" from the snippet "*Portland is the largest city* in Oregon. The skyline, seen here across the Willamette River…" However, the correct answer is "Berlin". This wrong answer is due to the fact that this answer pattern lacks the restriction of the question term "Germany"("Q_LCN"). As a rule, more complex answer pattern, i.e. including more question terms, is more valid to extract the correct answer. Thus it is necessary to evaluate these answer patterns.

The approach of answer pattern evaluation is as follows.

1. Query for each answer pattern of the question is formed and submitted to Google, and then the top 100 snippets are downloaded for answer pattern evaluation. The query consists of three parts:

   [Head]+[Tail]+[Q_Focus+Q_NameEntity]

   Where, [Head] stands for the string before the tag <A> of the answer pattern, and that [Tail] stands for the string after the tag <A> of the answer pattern. The value of them may be NULL, and [Q_Focus] or [Q_NameEntity] will be added into the query only if the [Head] and the [Tail] don't contain the term it represents. For the

above answer pattern "<A> Q_BeVerb Q_Focus" and sample question, the query is "is the largest city"+ "Germany". Here, [Head]= NULL, [Tail]= "is the largest city", [Q_Focus]= NULL and [Q_NameEntity] = "Germany"

2. The confidence of each answer pattern is calculated by the formula:

$$Confidence_P = Num_{Correct\_Match} / Num_{Match} \quad (4)$$

$Num_{Correct\_Match}$ denotes the number of snippets that tag <A> is matched by the correct answer, and Num_{Match} denotes the number of snippets that tag <A> is matched by any word.

3. At last the score of each answer pattern is computed as the formula: (λ=0.7)

$$Score_P = (1-\lambda) \bullet Weight_P + \lambda \bullet Confidence_P \quad (5)$$

Answer patterns with higher score lead to choose the answer with greater reliability, and those with lower score can't guarantee the correctness of its response. Some answer patterns along with their evaluation score are shown in Table 4.

The major advantage over other pattern based QA systems is that more than one question term can be included in the answer pattern, such as "Q_Focus in Q_LCN , <A>", containing two question terms "Q_Focus" and "Q_LCN". For longer question it is difficult to decide the unique question term containing the key information of the question, furthermore, the answer pattern containing more question terms is more confident for answer extraction.

Table 4. Sample Answer Pattern

Question Type	Answer Pattern	Score
[LCN]What Q_BeVerb Q_Focus in Q_LCN ? Sample question: What is the largest city in Germany ?	Q_Focus in Q_LCN Q_BeVerb <A> , <A> Q_BeVerb Q_Focus in Q_LCN Q_Focus in Q_LCN , <A> <A> Q_BeVerb Q_Focus	1.24 0.98 0.85 0.72
[DAT]When did Q_LCN Q_DoVerb Q_BNP ? Sample question: When did Hawaii become a state ?	Q_LCN Q_DoVerb Q_BNP in <A>. Q_DoVerb Q_BNP in <A>, Q_LCN in <A>, Q_LCN Q_DoVerb Q_BNP <A>, Q_LCN Q_DoVerb Q_BNP	0.98 0.86 0.86 0.77
[NUM]What Q_BeVerb Q_Focus of Q_BNP ? Sample question: What is the diameter of a golf ball ?	Q_Focus of Q_BNP Q_BeVerb <A>, Q_Focus of Q_BNP to <A>. Q_BNP Q_BeVerb <A> and Q_BNP Q_BeVerb <A>,	1.23 0.86 0.84 0.75
[PRN]Who Q_BeVerb Q_Focus of Q_LCN ? Sample question: Who is the prime minister of Australia ?	Q_Focus of Q_LCN <A> Q_BeVerb Q_BeVerb <A>, Q_Focus of Q_LCN Q_LCN Q_Focus <A> on <A> : Q_Focus of Q_LCN	1.54 0.98 0.91 0.82

4 Answer Extraction

Considering massive amounts of data on Internet, we select Google as the search engine for our question answering system. For each query submitted, Google returns top 100 snippets for answer extraction.

The answer patterns can be used to extract answer to a new question as follows:

(Sample question "What is the most populous city in the United States?" is used for explaining the following algorithm)

1. Identify the Q_Tag of the new question and then generate its Q_Pattern.
   Sample Q_Tag:
   Q_BeVerb="is" Q_Focus="the most populous city" Q_LCN= "the United States"
   Sample Q_Pattern: What Q_BeVerb Q_Focus in Q_LCN ?
2. Determining the question type of the question based on its Q_Pattern and answer type. The corresponding answer patterns of this question type are also selected from the predefined answer patterns.
   Sample question type: [LCN] What Q_BeVerb Q_Focus in Q_LCN ?
   Sample answer pattern: , <A> Q_BeVerb Q_Focus in Q_LCN
3. Replace Q_Tag symbols of each answer pattern with the corresponding question term of the question.
   Sample answer pattern is instantiated as:
   , <A> is the most populous city in the United States
4. For each answer pattern and each snippet returned, select the words matching tag <A> as the candidate answer.
5. Discard the candidate answers which don't satisfy the answer type of the question, using name entity tagger.
6. Sort the remainder candidate answers by their answer pattern's score and their frequency, and the one with the highest score is selected as the final answer.

In this stage, we resolve the problem of answer semantic restriction. As for the sample question, candidate answer "it" is extracted from the snippet "More than seven million people live in New York City*, it is the most populous city in the United States."*, using the answer pattern *", <A> Q_BeVerb Q_Focus in Q_LCN"*. However, the answer type of this question is "LCN", and this candidate answer doesn't satisfy this restriction then it is discarded.

5 Performance Analysis

We take the data of TREC-9 and TREC-10 as training examples for learning these answer patterns. To evaluate the performance of this approach we have done experiment, using the 500 questions of TREC-11.

The performance of QA system is influenced by the amount of text returned by the search engine. Fig.3 illustrates the impact of the retrieved snippet number, grouped by various interrogatives. The result is measured by the Mean Reciprocal Rank (MRR) score [7], a precision-like measure.

Here, *Num* denotes the maximum number of snippets search engine (Google) returned. The precision gets great improvement when *Num* is increased from 50 to 100 for more relevant snippets are returned, on the other hand, it doesn't increase any more when *Num* is increased to 200. When too many snippets are returned, the actually relevant snippets are submerged in a large amount of text, consequently a very large number of candidate answers are extracted and the system does not always rank the correct answer within the top five. Thus in our system the default maximum number of snippets is set to 100.

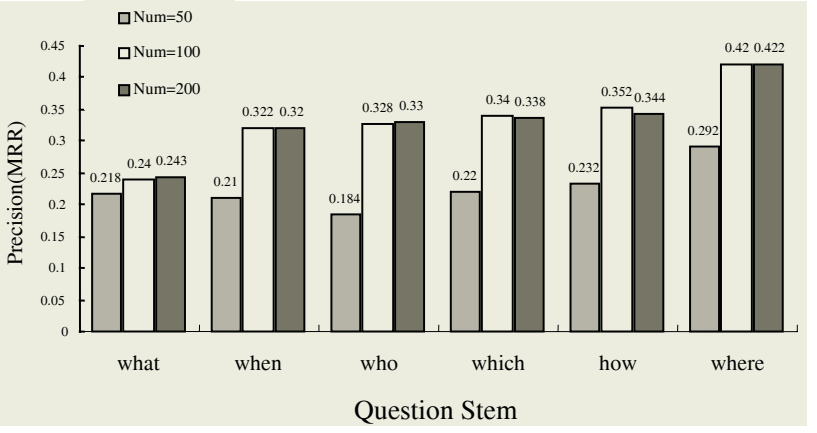

Fig. 3. Impact of maximum number of snippets processed

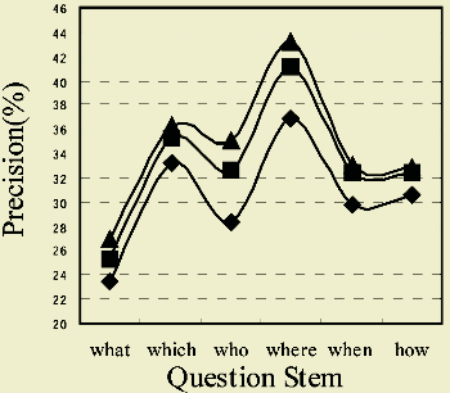

Fig. 4. Percentage of Correct Answer Across Various Question Stems

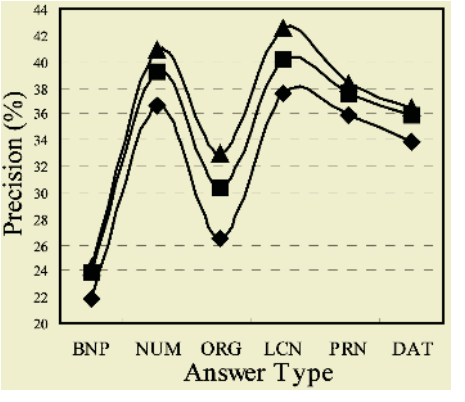

Fig. 5. Percentage of Correct Answer Across Various Answer Types

For some questions, the system doesn't return correct answer due to the bad ranking of candidate answers, in fact, correct answer has been extracted with the lower score. We analyze the result when different top k answers are considered for evaluation. Fig.4 and Fig.5 illustrate the experiment results across various interrogatives and answer types respectively.

We find a majority of correct answers are contained in the top 10 candidate answers and the performance gets greater improvement compared to the result of only top 1 answer is considered. It shows the shortcoming of our system that we should not only depend on the pattern score for candidate answer ranking but also other factors, such as the relevant degree of snippet to the question.

The overall precision of 500 questions is 0.309 and this result is within the top 1/3 groups in TREC-11 using the precision for evaluation.

6 Conclusion

The design of our QA system is a test for the novel pattern learning technology and the efficiency of this approach depends on the quantity and diversification of answer patterns largely. We take part in the TREC-12 this year and the primary evaluation result shows our result is above the median score of all runs submitted.

Among all these answer patterns what we have learned, some are too specific that they are almost useless for answering the new question. For instance, one of the answer patterns to the question "*What is a shaman?*" is "*Q_Focus was the priest, the <A> and*", where "Q_Focus" denotes the question term "*a shaman*". Here, "*the priest*" is related to this question closely and then this answer pattern is almost useless for answering new question. We will eliminate this kind of answer patterns in the future.

At present we only take the data of TREC-9 and TREC-10 as the training examples and that only top 100 documents to each <Q_Tag, Answer> pair query are downloaded for answer pattern learning, thus the number of answer patterns we have learned is restricted on account of the above factors, which influences the performance of our system. But the result is encouraging and we will go on with the development for higher performance. We believe it is an effective approach when more reliable answer patterns are learned.

Acknowledgements

This research was partly supported by NSF(Beijing) under contracts of 69935010 and 60103014, as well as the 863 National High-tech Promotion Project (Beijing) under contracts of 2001AA114120 and 2002AA142090.

References

1. Deepak Ravichandran and Eduard Hovy. 2002. Learning Surface Text Patterns for a Question Answering System. Proceedings of the ACL Conference.
2. Dell Zhang, Wee Sun Lee. 2002. Web based Pattern Mining and Matching Approach to Question Answering. Proceedings of the TREC-11 Conference. NIST, Gaithersburg, MD, 505-512.
3. M.M. Soubbotin. 2001. Patterns of Potential Answer Expressions as Clues to the Right Answer. Proceedings of the TREC-10 Conference. NIST, Gaithersburg, MD, 175-182.
4. Susan Dumais, Michele Banko, Eric Brill, Jimmy Lin, and Andrew Ng. 2002. Web Question Answering: Is More Always Better? Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 2002, Tampere, Finland.
5. E. Brill, J.Lin, M.Banko, S. Dumais, and A. Ng. 2001. Data-Intensive Question Answering. Proceedings of the TREC-10 Conference. NIST, Gaithersburg, MD, 183-189.
6. Cody C. T. Kwok, Oren Etzioni and Daniel S. Weld. May 1-5, 2001. Scaling Question Answering to the Web. Tenth World Wide Web Conference. Hong Kong, China.
7. Voorhees, E. 2001. Overview of the Question Answering Track. Proceedings of the TREC-10 Conference. NIST,Gathersburg, MD, 157-165.