ISSN:2249-5789

Extracting Answer in QA System: Learning Based

Megha Mishra[1], Vishnu Kumar Mishra[2], H.R. Sharma[3]

[1]*Research Scholar SOA University*
[2]*Asstt. Professor, BIT Durg*
[3]*Dean R&D, RECT Raipur*

[1]megha16shukla@gmail.com
[2]vshn _mshr@rediffmail.com
[3]hrsharmaji@indiatimes.com

***Abstract*— Converting questions to effective queries is crucial to open-domain question answering systems. In this paper, we present a web-based unsupervised learning approach for transforming a given natural-language question to an effective query. The method involves querying a search engine for web passages that contain the answer to the question, extracting patterns that characterize fine-grained classification for answers. Independent evaluation on a set of questions shows that the proposed approach outperforms a naive keyword based approach.**

***Keywords*— Question Answering, Machine Learning, Query Retrieval.**

I. INTRODUCTION

An automated *question answering* (QA) system receives a user's natural-language question and returns exact answers by analyzing the question and consulting a large text collection [1, 2]. As Moldovan et al. [3] pointed out; over 60% of the QA errors can be attributed to ineffective question processing, including query formulation and query expansion. A naive solution to query formulation is using the keywords in an input question as the query to a search engine. However, it is possible that the keywords may not appear in those answer passages which contain answers to the given question. For example, submitting the keywords in "*Who invented washing machine?*" to a search engine like Google may not lead to retrieval of answer passages like "*The inventor of the automatic washer was John Chamberlain.*" In fact, by expanding the keyword set ("*invented*", "*washing*", and "*machine*") with *"inventor of,"* the query to a search engine is effective in retrieving such answer passages as the top-ranking pages. Hence, if we can learn how to associate a set of questions (e.g. (*"who invented …?*") with effective keywords or phrases (e.g. *"inventor of"*) which are likely to appear in answer passages, the search engine will have a better chance of retrieving pages containing the answer. In this paper, we present a novel Web-based unsupervised learning approach to handling question analysis for QA systems. In our approach, training-data questions are first analyzed and classified into a set of fine-grained categories of question patterns. Then, the relationships between the question patterns and n-grams in answer passages are discovered by employing a word alignment technique. Finally, the best query transforms are derived by ranking the n-grams which are associated with a specific question pattern. At runtime, the keywords in a given question are extracted and the question is categorized. Then the keywords are expanded according the category of the question. The expanded query is the submitted to a search engine in order to bias the search engine to return passages that are more likely to contain answers to the question. Experimental results indicate the expanded query indeed outperforms the approach of directly using the keywords in the question.

II. RELATED WORK

Recent work in Question Answering has attempted to convert the original input question into a query that is more likely to retrieve the answers. Hovy et al. [2] utilized WordNet hypernyms and synonyms to expand queries to increase recall. Hildebrandt et al. [4] looked up in a pre-compiled knowledge base and a dictionary to expand a definition question. However, blindly expanding a word using its synonyms or dictionary gloss may cause undesirable effects. Furthermore, it is difficult to determine which of many related word senses should be considered when expanding the query. Radev et al. [5] proposed a probabilistic algorithm called *QASM* that learns the best query expansion from a natural language question. The query expansion takes the form of a series of operators, including INSERT, DELETE, REPLACE, etc., to paraphrase a factual question into the best search engine query by applying Expectation Maximization algorithm. On the other hand, Hermjakob et al. [6] described an experiment to observe and learn from human subjects who were given a question and asked to write queries which are

most effective in retrieving the answer to the question. First, several randomly selected questions are given to users to "manually" generate effective queries that can bias Web search engines to return answers. The questions, queries, and search results are then examined to derive seven query reformulation techniques that can be used to produce queries similar to the ones issued by human subjects. In a study closely related to our work, Agichtein et al. [7] presented *Tritus* system that automatically learns transforms of wh-phrases (e.g. expanding "*what is*" to "*refer to*") by using FAQ data. The wh-phrases are restricted to sequences of function word beginning with an interrogative, (i.e. who, what, when, where, why, and how). These wh-phrases tend to coarsely classify questions into a few types.*Tritus* uses heuristic rules and thresholds of term frequencies to learn transforms.In contrast to previous work, we rely on a mathematical model trained on a set of questions and answers to learn how to transform the question into an effective query. Transformations are learned based on a more fine-grained question classification involving the interrogative and one or more content words. Google search 2, originally developed by (Brin & Page, 1998), is a multilingual web search engine, indexing over one trillion unique web pages [10]. In this work, we use Google search to retrieve passages for factoid-type questions. These passages correspond to the summaries – also called *snippets* in IR jargon – that the search engine associates with each returned search result. Wikipedia 3 is a multilingual, web-based, and collaboratively edited encyclopaedia, with over thirteen million articles, as of 2009 [11]. DBpedia [12] is a *community effort to extract structured information from Wikipedia and to make this information available on the Web*. Put differently, DBpedia can be seen as a machine-readable version of Wikipedia, which provides a way to ask sophisticated queries against Wikipedia's unstructured contents, using declarative query languages, such as SPARQL.

III. Transforming Question Of Query

The method is aimed at automatically learning of the best transforms that turn a given natural language question into an effective query by using the Web as corpus. To that end, we first automatically obtain a collection of answer passages (*AP*s) as the training corpus from the Web by using a set of (*Q*, *A*) pairs. Then we identify the question pattern for each *Q* by using statistical and linguistic information. Here, a question pattern *Q*p is defined as a question word plus one or two keywords that are related to the question word. *Q*p represents the question intention and it can be treated as a preference indicative for fine-grained type of named entities. Finally, we decide the transforms *T*s for each *Q*p by choosing those phrases in the *AP*s that are statistically associated with *Q*p and adjacent to the answer *A*.

A. Search The Web for Relevant Answer Passages

For traning purpose, a large amount of question/answer passage pairs are mined from the Web by using a set of question/answer pairs as seeds. More formally, we attempt to retrieve a set of (*Q*, *AP*) pairs on the Web for training purpose, where *Q* stands for a natural language question, and *AP* is a passage containing at least one keyword in *Q* and *A* (the answer to *Q*). The seed data (*Q*, *A*)pairs can be acquired from many sources, including trivia game Websites, TREC QA Track benchmarks, and files of Frequently Asked Questions (FAQ). The output of this training-data gathering process is a large collection of (*Q*, *AP*) pairs. We describe the procedure in details as follows:

1.) For each (*Q*, *A*) pair, the keywords *k*1, *k*2… *k*n are extracted from *Q* by removing stop words.

2.) Submit (*k*1, *k*2,…, *k*n, *A*) as a query to a search engine *SE*.

3.) Download the top *n* summaries returned by *SE*.

4.) Separate sentences in the summaries, and remove HTML tags, URL, special character references (e.g., "& lt ;").

5.) Retain only those sentences which contain *A* and some *k*i.

B. Question Analysis

The subsection describes the presented identification of the so-called "question pattern" which is critical in categorizing a given question and transforming the question into a query. Formally, a "question pattern" for any question is defined as following form: *question-word head-word+* where "question-word" is one of the interrogatives (Who/What/Where/When/How) and the "head-word" represents the headwords in the subsequent chunks that tend to reflect the intended answer more precisely. If the first headword is a light verb, an additional headword is needed. For instance, "*who had hit*" is a reasonable question pattern for "*Who had a number one hit in 1984 with 'Hello'?*", while "*who had*" seems to be too coarse. In order to determine the appropriate question pattern for each question, we examined and analyzed a set of questions which are part-of-speech (POS) tagged and phrase-chunked. With the help of a set of simple heuristic rules based on POS and chunk information, fine-grained classification of questions can be carried out effectively.

Question Pattern Extraction

After analyzing recurring patterns and regularity in quizzes on the Web, we designed a simple procedure to recognize question patterns. The procedure is based on a small set of prioritized rules. The question word which is one of the wh-words ("*who*," "*what*," "*when*," "*where*," "*how*," or "*why*") tagged as determiner or adverbial question word. According to the result of POS tagging and phrase chunking, we further decide the main verb and the voice of the question. Then, we apply the following expanded rules to extract words to form question patterns:

Rule 1: *Question word in a chunk of length more than one).*
Qp = question word + headword in the same chunk
Rule 2: *Question word followed by a light verb and Noun Phrase (NP) or*
Prepositional Phrase (PP) chunk.
Qp = question word + light verb +headword in the following NP or PP chunk
Rule 3: *Question word followed immediately by a verb.*
Qp = question word + headword in the following Verb Phrase (VP) or NP chunk
Rule 4: *Question word followed by a passive VP.*
Qp = Question word + "to be" + headword in the passive VP chunk
Rule 5: *Question word followed by the copulate "to be" and an NP.*
Qp = Question word + "to be" + headword in the next NP chunk.
Rule 6: *If none of the above rules are applicable, the question pattern is the question word.*

By exploiting linguistic information of POS and chunks, we can easily form the question pattern. These heuristic rules are intuitive and easy to understand. Moreover, the fact that these patterns which tend to recur imply that they are general and it is easy to gather training data accordingly. These question patterns also indicate a preference for the answer to be classified with a fine-grained type of proper nouns. In the next section, we describe how we exploit these patterns to learn the best question to-query transforms.

C. Learning Best Transforms

This section describes the procedure for learning transforms *T*s which convert the question pattern *Q*p into bigrams in relevant *AP*s.

Word Alignment Across Q and AP

We use word alignment techniques developed for statistical machine translation to find out the association between question patterns in *Q* and bigrams in *AP*. The reason why we use bigrams in *AP*s instead of unigrams is that bigrams tend to have more unique meaning than single words and are more effective in retrieving relevant passages. We use Competitive Linking Algorithm [8] to align a set of (*Q*, *AP*) pairs. The method involves preprocessing steps for each (*Q*, *AP*) pair so as to filter useless information:

1.) Perform part-of-speech tagging on *Q* and *AP*.
2.) Replace all instances of *A* with the tag <ANS> in *AP*s to indicate the location of the answers.
3.) Identify the question pattern, *Q*p and keywords which are *not* a named entity. We denote the question pattern and keywords as *q*1, *q*2... *q*n.
4.) Convert *AP* into bigrams and eliminate bigrams with low term frequency (tf) or high document frequency (df). Bigrams composed of two function words are also removed, resulting in bigrams *a*1, *a*2... *a*m. We then align *q*'s and *a*'s via Competitive Linking Algorithm (CLA) procedure as follows:

Input: A collection *C* of (*Q*; *A*) pairs, where (*Q*; *A*) = (*q*1 = *Q*p , *q*2, *q*3... *q*n; *a*1, *a*2... *a*m)
Output: Best alignment counterpart *a*'s for all *q*'s in *C*

1. For each pair of (*Q*; *A*) in *C* and for all *q*i and *a*j in each pair of C, calculate LLR(*q*i,*a*j), logarithmic likelihood ratio (LLR) between *q*i and *a*j, which reflects their statistical association.
2. Discard (*q*, *a*) pairs with a LLR value lower than a threshold.

3. For each pair of (*Q*; *A*) in *C* and for all *q*i and *a*j therein, carry out Steps 4-7:

4. Sort list of (*q*i, *a*j) in each pair of (*Q; A*) by decreasing LLR value.
5. Go down the list and select a pair if it does not conflict with previous selection.
6. Stop when running out of pairs in the list.
7. Produce the list of aligned pairs for all *Q*s and *AP*s.
8. Tally the counts of aligning (*q*, *a*).
9. Select top *k* bigrams, *t*1, *t*2... *t*k, for every question pattern or keyword *q*.

The LLR statistics is generally effective in distinguishing
related terms from unrelated ones. However, if two terms occur frequently in questions, their alignment counterparts will also occur frequently, leading to erroneous alignment due to indirect association. CLA is designed to tackle the problem caused by indirect association. Therefore, if we only make use of the alignment counterpart of the question pattern, we can keep the question keywords in *Q* so as to reduce the errors caused by indirect association. For instance, the question "*How old was Bruce Lee when he died?*" Our goal is to learn the best transforms for the question pattern "*how old*." In other words, we want to find out what terms are associated with "*how old*" in the answer passages. However, if we consider the alignment counterparts of "*how old*" without considering those keyword like "*died*," we run the risk of getting "*died in*" or "*is dead*" rather than "*years old*" and "*age of*." If we have sufficient data for a specific question pattern like "*how long*," we will have more chances to obtain alignment counterparts that are effective terms for query expansion.

D. Runtime Transformations of Questions

At runtime ,a given question *Q* submitted by a user is converted into one or more keywords and a question pattern, which is subsequently expanded in to a sequence of query terms based on the transforms obtained at training. We follow the common practice of keyword selection in formulating *Q* into a query:

• Function words are identified and discarded.
• Proper nouns that are capitalized or quoted are treated as a single search term with quotes. Additionally, we expand the question patterns based on alignment and proximity considerations:
• The question pattern *Q*p is identified according to the rules (in Section 3.2) and is expanded to be a disjunction (sequence of ORs) of *Q*p's headword and *n* top-ranking bigrams (in section 3.3)
• The query will be a conjunction (sequence of ANDs) of expanded *Q*p, proper names, and remaining keywords. Except for the expanded *Q*p, all other proper names and keywords will be in the original order in the given question for the best results. For example, formulating a query for the question "*How old was Bruce Lee when he died?*" will result in a question pattern "*how old*." Because there is a proper noun "*Bruce Lee*" in the question and a remaining keyword "*died*," the query becomes "('*old*' OR '*age of*' OR '*years old*') AND '*Bruce Lee*' AND '*died*.'" Table 4 lists the query formulating for the example question.

IV. Experiment And Evaluation

The proposed method is implemented by using the Web search engine, Google, as the underlying information retrieval system. The experimental results are also justified with assessing the effectiveness of question classification and query expansion. We used a POS tagger and chunker to perform shallow parsing of the questions and answer passages. The tagger was developed using the Brown corpus and WordNet. The chunker is built from the shared CoNLL-2000 data provided by CoNLL-2000. The shared task CoNLL-2000 provides a set of training and test data for chunks. The chunker we used produces chunks with an average precision rate of about 94%.

A. Evaluation of Question Pattern

The 200 question from TREC-8 QA Track provide an independent evaluation of how well the proposed method works for question pattern extraction works. We will also give an error analysis.

TABLE I
THE FIREST FIVE QUESTIONS WITH QUESTION PATTERN AND JUGDUMENT

Question	Question Pattern	Judgment
Who is the author of the book," The Iron Lady: A Biography of Margaret Thatcher"?	Who-author	good
What was the monetary value of the Nobel Peace Prize in 1989?	What Value	good
What does the Peugeot Company manufacture?	What do manufacture	good
How much did Mercury spend on advertising in 1993?	How much	good
What is the name of the managing director of Apricot Computer?	What Name	bad

Two human judges both majoring in Foreign Languages were asked to assess the results of question pattern extraction and give a label to each extracted question pattern. A pattern will be judged as "good" if it clearly expresses the answer preference of the question; otherwise, it is tagged as "bad." The precision rate of extraction for these 200 questions is shown in Table 5. The second column indicates the precision rate when both of two judges agree that an extracted question pattern is "good." In addition, the third column indicates the rate of those question patterns that are found to be "good" by either judge. The results imply that the proposed pattern extraction rules are general, since they are effective even for questions independent of the training and development data. Table 6 shows evaluation results for "two 'good' labels" of the first five questions. We summarize the reasons behind these bad patterns:

• Incorrect part-of-speech tagging and chunking
• Imperative questions such as "*Name the first private citizen to fly in space.*"
• Question patterns that are not specific enough

For instance, the system produces "*what name*" for "*What is the name of the chronic neurological autoimmune disease which ...?*", while the judges suggested that "*what disease*." Indeed, some of the patterns extracted can be modified to meet the goal of being more fine-grained and indicative of a preference to a specific type of proper nouns or terminology.

V. CONCLUSION

In this paper, we introduce a method for learning query transformations that improves the ability to retrieve passages with answers using the Web as corpus. The method involves question classification and query transformations using a learning-based approach. We also describe the experiment with over 3,000 questions indicates that satisfactory results were achieved. The experimental results show that the proposed method provides effective query expansion that potentially can lead to performance improvement for a question answering system.

References

[1] Ittycheriah, A., Franz, M., Zhu, W.-J., and Rathaparkhi, A. 2000. IBM's statistical question answering system. In Proceedings of the TREC-9 Question Answering Track, Gaithersburg, Maryland.

[2] Hovy, E., Gerber, L., Hermjakob, U., Junk, M., and Lin, C.-Y. 2000. Question answering in Webclopedia. In Proceedings of the TREC-9 Question Answering Track, Gaithersburg, Maryland.

[3] Moldovan D., Pasca M., Harabagiu S., & Surdeanu M. 2002. Performance Issues and error Analysis in an Open-Domain Question Answering System. In Proceedings of the 40th Annual Meeting of ACL, Philadelphia, Pennsylvania.

[4] Hildebrandt, W., Katz, B., & Lin, J. 2004. Answering definition questions with multiple knowledge sources. In Proceedings of the 2004 Human Language Technology Conference and the North American Chapter of the Association for Computational.

[5] Radev, D. R., Qi, H., Zheng, Z., Blair-Goldensohn, S., Fan, Z. Z. W., and Prager, J. M. 2001. Mining the web for answers to natural language questions. In Proceedings of the International Conference on Knowledge Management (CIKM-2001), Atlanta, Georgia.

[6] Hermjakob, U., Echihabi, A., and Marcu, D. 2002. Natural Language Based Reformulation Resource and Web Exploitation for Question Answering. In Proceeding of TREC-2002, Gaithersburg, Maryland.

[7] Agichtein, E., Lawrence, S., and Gravano, L. Learning to find answers to questions on the Web. 2003. In ACM Transactions on Internet Technology (TOIT), 4(2):129-162. 8. Melamed, I. D. 1997. A Word-to-Word Model of Translational Equivalence. In Proceedings of the 35st Annual Meeting of ACL, Madrid, Spain.

[8] Yi-Chia Wang, Jian-Cheng Wu, Tyne Liang, and Jason S. Chang. 2004. Using the Web as Corpus for Un-supervised Learning in Question Answering, Proceedings of Rocling 2004, Taiwan.

[9] Google. (2008,July). *We knew the web was big* http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html.

[10] Wikipedia.(2009,October). *Wikipedia.* http://en.wikipedia.org/wiki/Wikipedia;

[11] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., & Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. In *In 6th int'l semantic web conference, busan, korea* (pp. 11–15). Springer.