Learning the Query Generation Patterns

Marcin Skowron and Kenji Araki

Graduate School of Information Science and Technology,
Hokkaido University, Kita-ku, Kita 14,
Nishi 9, Sapporo, Japan 060–0814
{ms, araki}@media.eng.hokudai.ac.jp

Abstract. With the current method of query formation, a Question Answering system retrieves a set of documents that are similar to a question, while what is mostly required is a set where an answer occurs frequently. This paper addresses this problem by presenting the Query Generation Pattern method. The aim of the method is to automatically learn an optimal combination, modifications of question words and possible extension of a query with non-question words, which for a given question category and syntax, form reliable queries that retrieve an answer-rich set of documents.

1 Introduction

In a Question Answering (QA) system, locating answers requires text analysis at a level of details that cannot be performed at a satisfactory retrieval time for large text collections[1]. As a result, most of the current QA systems employ a two-stage approach. The aim of the first stage is to select a set of documents relevant to a query from the whole document collection. In the second stage, a detailed analysis of a selected set is performed to find answers. Although, the performance of the second stage, and consequently of a whole QA system depends heavily on the quality of documents retrieved in the first stage, to date the research in this area drew relatively little attention.

The commonly used keyword based and similar approaches to a query formation do not retrieve an optimal set of documents. With this approach, a QA system retrieves a set of documents that are similar to a question, while what a user requests is an answer. In order to provide an answer, a QA system needs to retrieve a set of documents, where such answers occur frequently. In our opinion, for a given question category and question syntax, patterns that transform a given question into a reliable query can be automatically learned in a training process using question-answer pairs. Below, we introduce the idea and learning process of the Query Generation Pattern method. The preliminary test demonstrated a significant improvement in the results, compared to the commonly used keyword based and similar methods of query formation.

A. Gelbukh (Ed.): CICLing 2005, LNCS 3406, pp. 620–623, 2005.

2 Basic Idea of Query Generation Pattern

In the current QA systems, a query is often generated by removing the functional and stop words. For example, for the question "What does BBC stand for?", the query (BBC stand) or (BBC stands) would be formed. Once submitted to a search engine, this query retrieves a distorted set of documents where the correct answer - "British Broadcasting Corporation" occurs relatively infrequently. However, for the same question a more reliable query ("BBC stands for"), can be formed to retrieve a less distorted set of documents. To generate such a query, the preposition (excluded in the current method), and knowledge of which words to use, as well as how to modify (stand → stands) and order them to form an "exact phrase", is required. In the current approach, these means are not available. Moreover, the queries generated by the current QA systems do not provide any information on where to expect an answer candidate to appear, thus complicating the candidates' extraction process. Such information can be associated with the latter query ("BBC stands for" <answer candidate>). Additional improvement of query reliability can be achieved by extending it with a word or other non-letter characters, which frequently connect a given query with a correct answer. For example, for the question "When was Al Pacino born?", (question category: NUM:date) the query ("Al Pacino was born on") can be generated by the addition of the preposition "on", which is frequently found with the answer, like in the phrase "Al Pacino was born on 25 April 1940 [..]".

We think that for the questions from the same question category and with similar syntax, reliable queries are formed in a similar manner. These transformation patterns can be represented using the POS tags assigned to question words and by providing information on possible query extension with words that do not appear in an original question. For example, for the question "When was Queen Victoria born?" (syntax: /WRB1/VBD1/NNP1/NNP2/VBN1/?) (question category: NUM:date) a reliable query can be generated using the pattern: ("/NNP1 /NNP2 /VBD1 /VBN1 on") learned from the previous example from the same question category and with similar syntax " When was Al Pacino born?". This pattern forms the query ("Queen Victoria was born on"). The idea of a Query Generation Pattern (QGP) method[4] is to automatically learn an optimal combination, and modifications of question and non-question words, which for a given question category and syntax form a set of reliable queries. The aim of such a query is to retrieve a set of documents, where answers occur frequently and to indicate the possible localization of an answer candidate, which further simplifies the answer candidates extraction process.

The effectiveness of surface patterns was demonstrated in the system that best performed in TREC10 QA track[5]. This achievement resulted in further researches that described methods for automatic acquisitions of surface patterns and provided further evaluation of this method[1][3][6]. These works also revealed several shortcomings and limitations, like the fact that the patterns could include only one question key phrase; inability to handle more complicated question syntax, and very limited scope of question types. The QGP method described in this paper provides the means to learn question patterns automatically for the wide range of question types[1]. It differs also from the previous researches in extensive usage of information of syntax structure of questions and by combining several query formation techniques like “exact match”, question words modification and query extension with non-question words, into one complex query generation method.

3 Learning and Testing Query Generation Patterns

In the training process we used a set of 50 question-answer pairs from the TREC QA Collection, from various question categories. The process was started with a query that consisted of an answer and question words including nouns, adjectives, adverbs, and verbs that were not on the stop-word list. Additionally, an initial query was extended with question related words, including various verb and noun forms derived from the main verb found in a question. For example, for the question “When did the Vesuvius last erupt?”, the query (1944+Vesuvius+last+erupt OR erupts OR erupted OR eruption OR erupting) was generated. From the set of 100 documents accessed with this query, sentences that contain an answer and at least one question or question related word were extracted. Using these sentences as training data, the list of most frequent n-grams that contained only the question and question related words was generated. For the example question, the list of the n-grams with an occurrence greater than the set threshold included strings like: “last erupted”, “last eruption”, and “the last eruption”. In the next step, discovered n-grams were extended with the remaining question words that did not appear in a given n-gram string, either directly or as one of the derivative forms. These constituted a set of the Query Generation Pattern (QGP) candidates. The reliability of a given candidate was calculated as a number of answers (the multiple occurrences of an answer in one snippet is counted only once) to the number of accessed snippets (a number between 1-100). The QGPs with the highest reliability score were selected and stored. Table 1 presents the results for some of the discovered patterns. The most reliable query found for this question is approximately 40% more reliable than one that could be generated by a current QA System (position 3) by extracting the keywords and transforming the verb form ((did) erupt → erupted). For all the questions from the training set, QGP method was able to discover queries more reliably than those formed using a keyword based approach.

In the same process, using all snippets containing a correct answer, the words that frequently link a question word with an answer - Connection Patterns (CP) - were discovered. The most frequent CPs were joined to the corresponding QGP. Such extended QGPs are verified using the method described above. If found to form a reliable query, it was added as an additional pattern to be stored along with a particular question category and question syntax. Table 1, position 1 presents the QGPs extended with the CP.

[1] Question types used in the training process included 50 fine-grained categories. For the details see http://l2r.cs.uiuc.edu/c̃ogcomp/Data/QA/QC/definition.html.

Table 1. Examples of QGP found for the question "When did the Vesuvius last erupt", (question category: NUM:date) (syntax: /WRB1/VBD1/DT1/NNP1/JJ1/VB1/?)

No.	Query	Query Pattern	Reliability Score
1	"last erupted in" Vesuvius	"/JJ1 /VB1_ed in" /NNP1	60
2	"last erupted" Vesuvius	"/JJ1 /VB1_ed" /NNP1	46
3	last erupted Vesuvius	/JJ1 /VB1_ed /NNP1	42
4	last eruption Vesuvius	/JJ1 /VB1_tion /NNP1	40
5	"the last eruption" Vesuvius	"/DT1 /JJ1 /VB1_tion" /NNP1	38

The application of the discovered QGPs for the set of 50 test questions (various question categories, syntax similar to the questions used in the training process) confirmed that using the learned patterns, the system could automatically generate a set of highly reliable queries that retrieved a set of documents where an answer occurred more frequently, compared to the currently used keyword based approach. For the test set, the improvement rate varied depending on the question, between 17%-76%.

4 Conclusions and Future Work

The Query Generation Pattern method demonstrates that the QA system can automatically acquire knowledge on how to form a set of reliable queries for a given question category and question syntax. Using this method, the system also obtains information on where an answer candidate is likely to occur and what words or non-letter characters frequently connect it to a given query, even if these elements were not present in a question. The preliminary results are promising, showing a significant improvement over the currently used method. Our future work includes extensive evaluation of the proposed method and providing the means to extend a query with words semantically related to a question.

References

1. Greenwood M. (2002) Question Answering. PhD Progress Report, http://www.dcs.shef.ac.uk/ mark/phd/work.
2. Hovy E. Hermajakob U., Ravichandran D. (2002) Proceedings of the Human Language Technology (HLT) Conference.
3. Ravichandran D., Hovy E. (2002) Learning Surface Text Patterns for a Question Answering System. In Proc. of 40th Annual Meeting of the Association for Computational Linguistics (ACL).
4. Skowron M., Araki K. (2003) Basic Idea of Corpus-Supported Approach to Question Answering. Convention Record of the Hokkaido Chapters of the IEEE.
5. Soubbotin M.M., Soubbotin S.M. (2002) Patterns of Potential Answer Expressions as Clues to the Right Answer. In Proc. of 10th Text retrieval Conference (TREC10).
6. Zhang D., Lee W. (2002) Web Based Pattern Mining and Matching Approach to Question Answering. In Proceedings of the 11th Text REtrieval Conference (TREC).