AN AUTOMATIC ANSWERING SYSTEM WITH TEMPLATE MATCHING

T. Gunawardena, M. Lokuhetti, N. Pathirana, R. G. Ragel and D. S. Deegalla

Department of Computer Engineering, Faculty of Engineering, University of Peradeniya

Introduction

Accessing information contained within large information sources in a simple and effective way is an issue gaining in importance in everyday life. Our system targets on providing timely information automatically to natural language questions. The approach adopted is an automated FAQ (Frequently Asked Question) answering system that provides pre-stored answers to user questions asked in ordinary English, rather than forcing the users to ask questions in a restricted syntax. A natural language processing technique that is developed for FAQ retrieval analysis is applied to FAQs in the database. Thus, the work of FAQ retrieval is reduced to keyword matching creating an illusion of intelligence. The system is developed such that it handles spellings and grammar mistakes in the question. It is both evolving and flexible QA systems are a hot topic of research at present. They allow users to ask questions in natural language and give a concise and accurate answer. Being a very active area of research and development, the NLP (Natural Language Processing) plays a big role in QA systems. NLP is a computerized approach to analyze text based on both a set of theories and a set of technologies. It will become important to be able to ask queries and obtain answers, using natural language expressions, rather than the keyword based retrieval mechanisms. So the QA system can better satisfy the needs of users or retrieval and find the answer the users need accurately in a quicker, more convenient and more effective way. There are two types of QA systems (1) closed-domain question answering deals with questions under a specific domain (2) open-domain question answering deals with questions about nearly everything.

Methodology

There are a number of existing methods (Kerdprasop *et al.*, 2008) for coming up with an appropriate answer for a user question. Fig. 1 categorizes these methods.

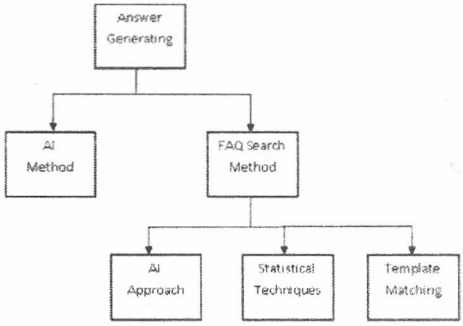

Fig. 1: Answer obtaining method

A. AI Method

The AI method (Palme, 2008) focuses on answer generation by analyzing questions and creating an "understanding" of the question. This requires complex and advanced linguistic analysis programs.

B. FAQ Search Method

There are three generic methods that an answer can be generated using stored FAQs and answers.

1. Artificial Intelligence Approach: This method uses an ontology-based knowledge base in order to comprehend the user question and then query the FAQ database.
2. Statistical techniques: Consider properties of the user question to decide whether it is equivalent to a FAQ. This approach works rather poorly for short questions and when the query and FAQ use different wording to carry the same meaning.
3. Template Matching: This is based on manually specifying templates for each Frequently Asked Question. The templates are matched against the questions asked by users. The success of the question answering thus depends a lot on the quality of these templates.

Template matching method has some advantages: (1) precision of the retrieval is high because the keywords are selected using human intelligence; (2) an understanding of the problem domain is not required for developing; and (3) system is both evolving and flexible. Evolving because its question answering ability improves as more questions are asked and new FAQ entries are created. It is flexible because the system could be used for any problem domain by simply changing the *knowledge base*.

The need of writing templates manually for all the questions is less efficient to the system.

Implementation and Testing

The implementation technique we have used here is the FAQ Search Method with template matching. The details of the template used and an example implementation are described below.

Frequently asked questions paired with their relevant answers are stored in a database after they are converted to templates. Syntax of the templates is defined so that a single template could match many variants of the same question. The different ways a question might be asked is due to: different tenses, singular/plural forms, synonyms, word order and optional words. The syntax used for making the templates is shown in following

Syntax used for generating FAQ templates

; (semicolon) - Used to separates terms. A question must contain all terms of a template in order to be considered a match.

/ (forward slash) - When words are separated by / either one of the words must match with the user question

* *(asterisk)* - This symbol at the end of a group of characters means that additional characters could follow.

Examples: go=going, gone, goes*
robo=robos, robot, robots, robotics*

[] (square parentheses) - Words grouped with [] denotes phrases.

: (colon) - Terms separated by a ":" should directly follow each other.

 # (hash) - Terms separated by hash, should appear in the designated order without necessarily being adjacent.

' ' (space) - Appears only within square parenthesis. Terms separated by spaces denotes a choice.

$ (dollar) - A '$' at the beginning of a terms specifies checking with the synonym list.
Disemvoweling (Maxwell, 2007) - Performed on templates as a measure of accounting for the spelling mistakes in user queries.

Using the above syntax arbitrary complex templates can be constructed. Also phrases can be nested within each others, and synonym list could also contain phrases that have the same meaning as a single word.

When a question is received by the system, it is matched against each and every template in the database until it finds the best matched template. The success of answering thus depends a lot on the quality of these templates. Further in this process, some words those are considered to have synonyms are referred in a synonym file so as to cover different variants of a question. Figure 2 shows how a template is improved to accommodate different variations of the same question. It is worth noting that we do not check for grammar correctness in the questions.

Therefore the final version of the template that is inserted to the database after disemvoweling is whr*/[hw#$g]/lctn;[cmptr*:$dprtmnt]. The one before disemvoweling is where*/[how#$go]/location;[computer*:$department].

The system was deployed and tested in an exhibition environment. The experimented results for a random set of questions concluded that the system can respond with accurate answers to the questions with a probability of 91%.

Where is the computer department?
Computer department where?
Where;[computer:department]

Where is the computer section?
Where;[computer:(department section)]
Where;[computer:$department)]

How can I go to the computer department?
Where[how go]; [computer:$department)]

How can I get to the computer department?
Where[how#(go get)];[computer:$department]
Where[how#$go];[computer:$department]

Fig. 2. An example template

Conclusion

The final result is a smart, user friendly automatic answering system with the ability of answering natural language questions.

References

Kerdprasop N., Pannurat N. and Kerdprasop, K. (2008). Intelligent Query Answering with Virtual, Mining and Materialized Views. World Academy of Science, Engineering and Technology, 48: 84-85.

Maxwell, K. (2007). Disemvowelling. Retrieved from http://www.macmillandictionary.com/buzzword/entries/disemvowelling.html

Palme, J. (2008). Principles of Intelligent Natural Language Question Answering. Retrieved from http://web4health.info/en/answers/project-search.htm