An Automatic Answering System with Template Matching for Natural Language Questions

Tilani Gunawardena, Medhavi Lokuhetti, Nishara Pathirana, Roshan Ragel and Sampath Deegalla

Faculty of Engineering, University of Peradeniya, Peradeniya 20400 Sri Lanka

etilani@gmail.com, medhavimpl@gmail.com, nishara.pdn@gmail.com, roshanr@pdn.ac.lk and dsdeegalla@pdn.ac.lk

Abstract— **Using computers to answer natural language questions is an interesting and challenging problem. Generally such problems are handled under two categories: open domain problems and close domain problems. This paper presents a system that attempts to solve close domain problems. Typically, in a close domain, answers to questions are not available in the public domain and therefore they cannot be searched using a search engine. Hence answers have to be stored in a database by a domain expert. Then, the challenge is to understand the natural language question so that the solution could be matched to the respective answer in the database. We use a template matching technique to perform this matching. In addition, given that our target is to use this system with non-native English speakers, we developed a method to overcome the mismatches we might encounter due to spelling mistakes. The system is developed such that the questions can be asked using short messages from a mobile phone and therefore the system is designed to understand SMS language in addition to English. One of the main contributions of this paper is the outcome presented of a deployment of this system in a real environment.**

Keywords—**FAQ, Answering System, SMS, Template Matching**

I. Introduction

DEVELOPING mechanisms for using computers to answer user questions is becoming an interesting problem with the increased use of computers. Such mechanisms allow users to ask questions in a natural language and give a concise and accurate answer. Understanding user questions in natural languages requires Natural Language Processing (NLP). Being an active area of research, NLP plays a big role in the ICT and Question Answering (QA) systems.

Natural language processing is the computerized approach to analyzing 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 (NL) expressions, rather than the keyword based retrieval mechanisms. The QA system can better satisfy the needs of users as they will provide an accurate, quicker, convenient and effective way of giving answers to user questions.

The approach we have adopted in this project is an automated FAQ (Frequently Asked Question) answering system that replies with pre-stored answers to user questions asked in ordinary English, rather than keyword or syntax based retrieval mechanisms. This is achieved using a template matching technique with some other mechanisms like disemvoweling, matching synonyms, etc.

The natural language processing technique developed for FAQ retrieval does not analyze user queries. Instead 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 both evolving and portable. Evolving because its question answering ability improves as more questions are asked and new FAQ entries are created. It is portable because the system could be used for any problem domain (closed) by changing the knowledge base.

Typically, there are two types of question answering systems: (1) closed-domain question answering that deals with questions under a specific domain, and can be seen as an easier task on one hand as the NLP systems can exploit domain-specific knowledge frequently formalized in ontology but harder on the other as the information is not generally available in the public domain; and (2) open-domain question answering that deals with questions about nearly everything, and can rely only on general ontology and world knowledge. On the other hand, as mentioned earlier these systems usually have much more data available in the public domain from which to extract the answer.

As depicted in Figure 1, there exist two methods [1], [2] for coming up with an appropriate answer for a user question and they are AI method and FAQ search method.

The AI method [2] focuses on answer generation by analyzing questions and creating an “understanding” of the question. This requires complex and advanced linguistic analysis programs. There are three generic methods that an answer can be generated using stored FAQs and answers [3] and they are: (1) artificial intelligence approach; (2) statistical techniques; and (3) template matching.

978-1-4244-8551-2/10/$26.00 ©2010 IEEE

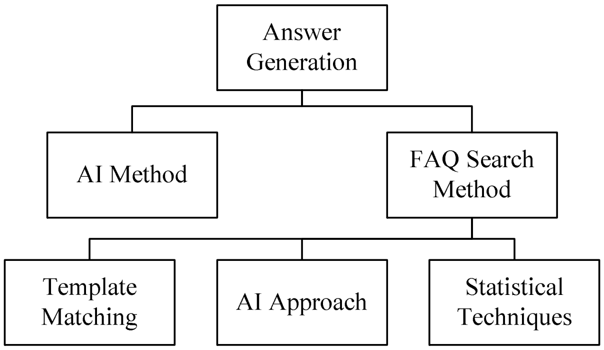

Fig. 1. Different methods for obtaining answers

The artificial intelligence approach uses an ontology-based knowledge base in order to comprehend the user question and then query the FAQ database. The statistical techniques consider the similarities in work, sentence length, word order or distance of identical work of the user question to decide whether it is equivalent to an FAQ. Values are assigned to reflect the similarity and then the sum of these values is used for comparison. This approach works rather poorly for short questions and when the query and FAQ use different wording to carry the same meaning. A closed domain question answering system based on the FAQ search method coupled with template matching has been adapted to service users in our system.

In the era of Information and Communications Technology (ICT), mobile phone has become a fast and convenient way to communicate over a network. Therefore, our system is further extended to mobile devices where the questions can be asked as short messages using SMS (Short Message Service). Through this extension we enable end users to access information regardless of their location, and at the time that is most convenient to them.

The rest of the paper is organized as follows. In Section II we describe the related work and the system architecture in Section III. In Section IV, we describe the template matching algorithm in detail and in Section V we have identified and used a number of techniques to enhance the template matching method. In Section VI, we detail the SMS interface that we have integrated into our system and Section VII is on a case study performed on our system as an exhibition information system. In Section VIII we analyse the system qualitatively and in Section IX we conclude the paper.

II. Related work

Q&A system research received considerable attention from the research community through Text Retrieval Conference [8] Q&A track since 1999. The original aim of the track is to systematically evaluate both academic and commercial Q&A systems. Maybury [9] has discussed the characteristics of Q&A systems and resources needed to develop and evaluate such systems. Main approaches in Q&A systems could be found in [10] in which template-based approach discussed in detail. Although, most Q&A systems are based on Web environments, SMS has also been used as an environment in contexts such as in learning [11] and agriculture [12].

III. System Architecture

In this section we describe the architecture of our system. The overall architecture of the system can be subdivided into three main modules: (1) pre-processing, (2) question template matching, and (3) answering. Figure 2 shows the system architecture of the question and answering system. Each module is described in detail in the following subsections.

A. Pre-Processing Module

Pre-processing module mainly consists of three operations: (1) converting SMS abbreviations into general English words, (2) removing stop words, and (3) removing vowels. Since the system is expected to process texts with both natural and SMS languages it is necessary to replace the SMS abbreviations with the corresponding English words before processing user questions further. This is done by referring to pre-stored frequently used SMS abbreviations. Stop words are the words that add no effect to the meaning of a sentence even if they are removed.

Removing stop words is done to increase the effectiveness of the system by saving time and disk space. Examples of stop words are the, a, and, etc. Next step in this module is to remove vowels from the text to handle spelling mistakes. This process is called disemvoweling which will be discussed in details in coming sections.

B. Question-Template Matching Module

The pre-processed text is matched against each and every pre stored template until it finds the best matched template with the received text. In order to do this, templates are created according to a specific syntax and the details are described in section IV. Further in this module, words that are considered to have synonyms are referred in a synonym file. This synonym file can be modified according to the relevant domain and are updated from a standard database such as WordNet [6]. It is worth noting that the templates here are for questions and not for answers. The main target of this system is to identify the closest template that matches the question we have received from the user.

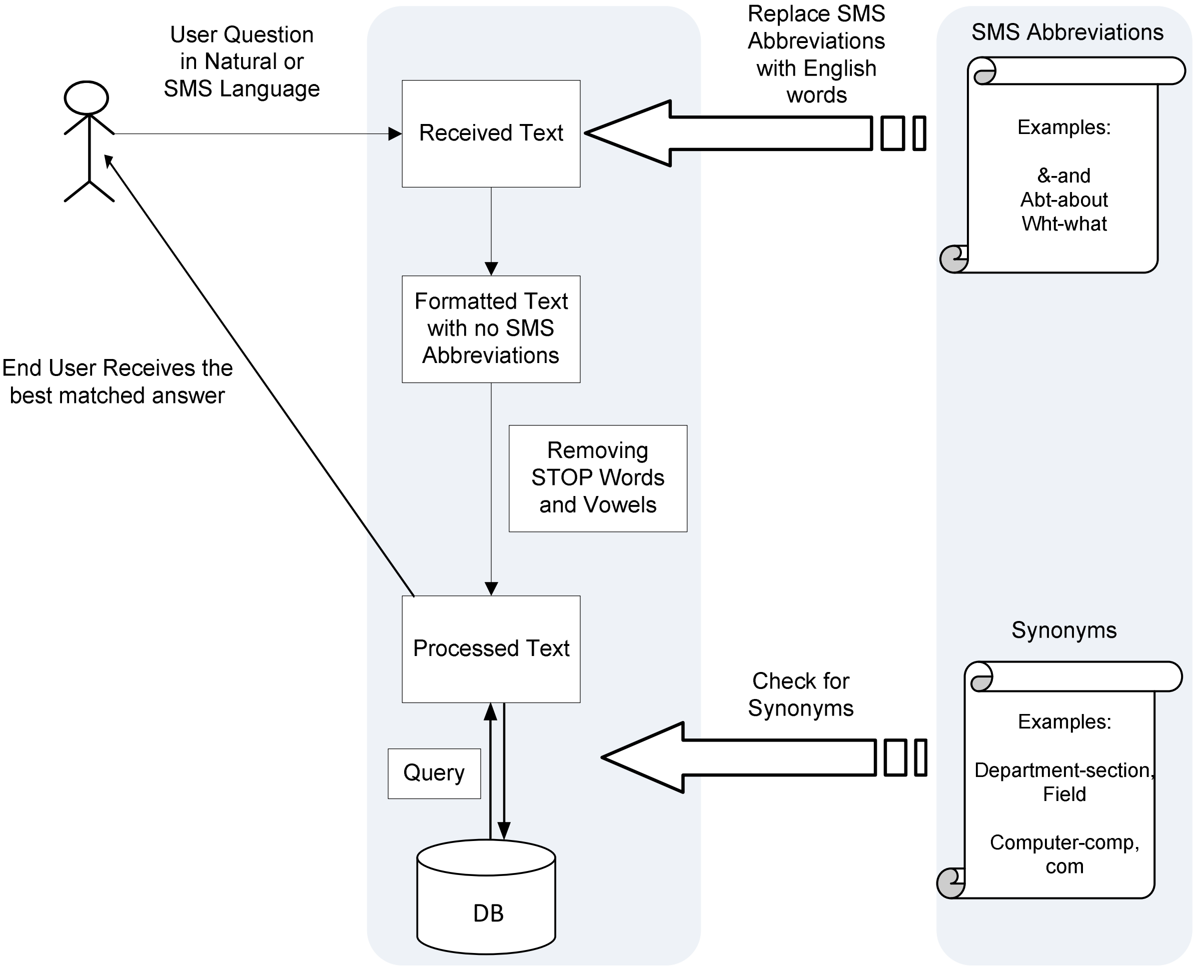

Fig. 2. System architecture of the question answering system

C. *Answering Module*

Since each and every template representing a question are pre stored in a database with its answer, just when the best matched template for the question is found, the corresponding answer will be returned to the end user.

IV. Template Matching

As mentioned earlier, the user questions are answered using template matching. In this section, we discuss the templates used and their syntax. Our method is based on manually specifying templates for each Frequently Asked Question. Those are stored in a database coupled with the answers. The templates are matched against the questions asked by users to find the best matched template. The success of the question answering thus depends a lot on the quality of these templates.

The syntax of the templates is defined so that a single template could match many different variants of the same question. A question might be asked in different ways due to one or more of the following reasons: different tenses; singular/plural forms; usage of synonyms; the order of using words; and the use of optional words.

In Table I, we tabulate the syntax used for the templates of the questions. Using the above syntax arbitrary complex templates can be constructed. Also phrases can be nested within each other, and synonym list could also contain phrases that have the same meaning as a single word.

TABLE I
THE SYNTAX USED FOR THE TEMPLATES

Syntax	Description
;	Used to separates terms. A question must contain all terms of a template in order to be considered a match.
/	When words are separated by / either one of the words must match with the user question.
*	This symbol at the end of a group of characters means that additional characters could follow. Used to handle stemming (reducing derived words to their base form) Examples: go* = going, gone, goes robo* = robos, robot, robots, robotics
[ ]	Words grouped with [] denotes phrases.
:	Used only within square parentheses. Terms separated by a ":" should directly follow each other.
#	Used only within square parentheses. Terms separated by hash, should appear in the designated order without necessarily being adjacent.
’ ’	Appears only within square parenthesis. Terms separated by spaces denotes a choice.
$	A '$' at the beginning of a terms specifies checking with the synonym list.

Following are identified as the advantages of using a template matching approach: (1) Precision of the retrieval is high because the keywords are selected using human intelligence; (2) It is an evolving system, because its question answering ability improves as more questions are asked, and new FAQ entries are added to the database; and (3) An understanding of the problem domain is not required for developing. The main disadvantage of the system is that the templates need to be written manually for all questions.

V. ENHANCING TEMPLATE MATCHING

The template matching technique is enhanced using two additional techniques and they are: (1) applying disemvoweling and (2) using a synonym list.

It is believed that most of the spelling mistakes occur because of omission, addition or out of order vowels. Therefore, removing vowels in a sentence will reduce the amount of spelling mistakes encountered in a sentence. Therefore vowels are removed from user questions in our system. The process of removing vowels in text is known as disemvoweling [4]. Disemvoweling is also done in our templates as a means of accounting for spelling mistakes in user queries and for easy matching of the templates. We believe that this is a vital addition to the system as the system is expected to be used by non-native English speakers who are prone to make ample spelling mistakes in their questions.

It is important to list synonyms for each term, since users potentially might often query using different terminology than the person who produces the FAQ. If the same list of synonyms occurs in many FAQs, it is put in a separate synonym list, and stored in a text file which is mapped into a Hash Map when the program loads. This list is referred when a '$' sign appears in a template. Example: $describe = describ* depict* illustr* specif* character* clarif*. To improve the quality of the synonyms list, we also have identified the usage of WordNet [6] through which we can expand our query terms.

VI. THE SMS INTERFACE

Short Messages (SM) as a communication medium has evolved so much as to be recognized as a language of its own through the efficient use of abbreviations coupled with the minimum use of characters in coming up with a comprehensible expression [7]. Given that we have given the facility of using SMs to ask questions, we expected the user to ask questions not only using natural language (English) but also using SMS language. Therefore it is necessary to replace the SMS abbreviations with the corresponding English words before further processing user questions and performing template matching. Replacing SMS abbreviations is performed by referring to a compilation of frequently used SMS abbreviations.

VII. A CASE STUDY- ENGEX2010

We have deployed and tested our system in an exhibition environment [5] which is a closed domain QA system. In the deployment users were allowed to send short messages with a question or a comment in natural language or SMS language. Since the exhibition was heavily crowded and the environment was unfamiliar to exhibition visitors, this information system allowed them to easily obtain both static and dynamic information without much overhead by simply using their mobile phones and SMS facility.

A survey for the type of questions expected and a careful study of the domain were used to select questions and the relevant answers for the deployment. Questions were converted to templates and were stored in the backend database. A trail period of testing and training was used to improve the templates and a large number of templates were developed and inserted with answers to the deployment. The particular exhibition [5] ran for seven days and the system was used by thousands of visitors.

We tabulate a sample template and a number of questions matched (and therefore answered) using that particular template. The final template inserted to the database after disemvoweling is **whr/[hw#$g]/lctn;[cmptr;$dprtmnt].** Therefore, the one that is before disemvoweling is **where/[how#$go]/location;[computer:$department]**. In the SMS abbreviation file we have used **dep = department dept = department, deptmnt = department** and in synonym file have used file **g = gt, rch, fnd ($go = get, reach, find) and dprtmnt = dprtmnt*, sctn*, bldng*,**

rm*, lb*, fld* ($department = department*, section*, bilding*, room*, lab*, field*). Therefore a few of the expected possible questions are like “where is the computer department”, “how to go to computer dept”, etc. As you can see from Table II, there are a number of other interesting questions that are matched to the template with the number of keywords matched.

TABLE II
SAMPLE TEMPLATE AND MATCHED USER QUESTIONS I

Template**: whr/[hw#$g]/lctn;[cmptr:$dprtmnt]**	
User questions	**keywords matched**
whr z com dept	2/2
whr is computer department	2/2
whr s computer dept	2/2
how to go to com department	2/2
how to find com dept	2/2
location of computer department	2/2
how to go com dept	2/2
how to reach com department	2/2
hw to get to com dept	2/2
com department where?	2/2

TABLE III
SAMPLE TEMPLATE AND MATCHED USER QUESTIONS II

Template**: whn/tm/dy;xhbtn/ngx;strt*/nd*/hld/pn***	
User questions	**keyword matched**
when exhibition start	3/3
when z exhibition start	3/3
whn s exhibition end	3/3
day of exhibition	2/3
time of exhibition	2/3
day exhbn start	3/3
when is exhtn held	3/3
when is the EngEx	2/3
when start Engex	3/3
when z engex end?	3/3

Another example of a template used during EngEx2010 is given in Table III. The template used in this occasion is **whn/tm/dy;xhbtn/ngx;strt*/nd*/hld/pn***. Therefore the template before disemvoweling is, **when/time/day;exhibition/engex;start*/end*/held/open*** and a couple of expected typical questions are “when will the exhibition start” and “time of the EngEx opening”. Please note EngEx is the name given to this particular exhibition. Table III also tabulates a number of user questions asked and matched this particular template and the number of keywords matched.

In Table IV, we tabulate the correct and incorrect answers given by our system for a set of 100 random questions asked by the visitors. We failed to find error rates of similar closed domain answering systems in the literature and therefore we are unable to perform a quantitative comparison of our system against them.

TABLE IV
RESULTS OF A RANDOM SET OF 100 QUESTIONS

% of keyword matched	**# of answers**	
	correct	**incorrect**
100%	72	02
50-99%	15	03
< 50%	04	04
Total	**91**	**09**

VIII. A QUALITATIVE ANALYSIS OF THE SYSTEM

Although the system worked according to our expectation, it cannot be guaranteed that the QA system will always provide accurate answers due to the following reasons.

A. Lack of understanding of the problem domain

Since we are using a compilation of FAQs, an understanding of the problem domain is a must for the preparation of the templates for the FAQs. In order to get a better understanding of the type of questions we could encounter, we have also conducted a survey among school children about the questions they might have period to the exhibition. But still we cannot expect all questions to be in accordance with the FAQs in the database.

If a received question is new to the existing database, the database can be updated with a new question and answer pair. Therefore the accuracy of the system is expected to be increased over time since the knowledge of the problem domain increases with usage.

B. Handling SMS abbreviations

The method we have used has many shortcomings because SMS abbreviations do not always obey or follow standard grammar, and the interpretations of abbreviated words largely depends on the context.

C. Handling spelling mistakes

Disemvoweling only has the ability to handle some of the spelling mistakes (where the vowels are used incorrectly). But since we are not getting feedbacks from the user, it is not possible to use a more advanced functionality such as a spell checker.

IX. CONCLUSION AND FUTURE WORK

The final result is a smart, user friendly automatic answering system with the ability of detecting and answering questions asked in English or SMS language. We propose to explore the shortcomings we discussed in Section VIII as future work.

ACKNOWLEDGMENT

We would like to thank all staff members of the Department of Computer Engineering at Peradeniya who

helped us whenever needed and the reviewers for their valuable feedback on the manuscript.

REFERENCES

[1] N. Kerdprasop, N. Pannurat and K. Kerdprasop, “Intelligent Query Answering with Virtual, Mining and Materialized Views”. World Academy of Science*, Engineering and Technology*, 48: pp: 84 – 85, 2008.

[2] J. Palme (2008, August 22). “Web 4 Health” [Online]. Available: http://web4health.info/en/answers/project-search.htm

J. Palme and E. Sneiders, “Natural Language Question Answering System Classification Manual”, REPORT D2.2-part B Revision 1.0, 2008.

[4] Maxwell, Kerry (2007, August 13). “Disemvowelling or disemvoweling” [Online]. *Word of the Week Archive*. Macmillan Available: http://www.macmillandictionaries.com/wordoftheweek/archive/070813-disemvowelling.htm

[5] Engineering Exhibition 2010 [Online]. Available: http://www.pdn.ac.lk/eng/EngEx2010

[6] Fellbaum, Ch. (ed). 1998. *WordNet: An Electronic Lexical Database*. Cambridge: MIT Press.

[7] Wikipedia Contributors, “The SMS Language” [Online]. Available: http://en.wikipedia.org/wiki/SMS_language

[8] *Text retrieval conference* [Online]. Available: http://trec.nist.gov

[9] M.T. Maybury, 2002, “Toward a Question Answering Roadmap”.

[10] A. Andrenucci and E. Sneiders, “Automated Question Answering”: Review of the Main Approaches, *in International Conference on Information Technology and Applications*, pp. 514-519, 2005.

[11] S.R. Balasundaram and B. Ramadoss. “SMS for Question-Answering in the m-Learning Scenario”, *Journal of Computer Science* 3 (2): pp. 119-121, 2007.

[12] M. Suktarachan, P. Rattnamanee and A. Kawtrakul, “The Development of a Question-Answering Services System for the Farmer through SMS: Query Analysis”, *in Proceedings of the 2009 Workshop on Knowledge and Reasoning for Answering Questions*, pp. 3-10, Suntec, Singapore, 2009.