ISSN:0975-9646

Automatic Answering System for English Language Questions

Abhay Mone , Ishwar Mete , Priyanka Gangarde ,Malhari Kharad

Department of Computer Engineering, P.E.S. Modern College of Engineering, Pune University, India

Abstract **—To answer English language questions using computer 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 both close and open domain problems. Answers to questions from close domain 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 English 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. 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- Answering System, SMS, Template Matching, JSON, Natural Language Processing.

I. INTRODUCTION

The goal of the Natural Language Processing is to design and build methods that will analyse, understand, and generate languages that humans use naturally, so that eventually we will be able to address our computer as though we were addressing another person.Understanding user questions in English languages requires Natural Language Processing.

[1]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 expressions, rather than the keyword based retrieval mechanisms. The question answering system can better satisfy the needs of users as they will provide an accurate and effective way of giving answers to user questions.

[1]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.

II. 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) Answer Discovery, and (3) Answering. Figure 1 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 two operations:

1. Converting SMS abbreviations into general English words.

2. Removing stop words and stem word s.

[1]The system is expected to process texts with both English 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 and Stem words are the words that add no effect to the meaning of a sentence, are removed. Stemming is a pre-processing step in Text Mining applications as well as a very common requirement of Natural Language processing functions. In fact it is very important in most of the Information Retrieval systems. The main purpose of stemming is to reduce different grammatical forms / word forms of a word like its noun, adjective, verb, adverb etc. to its root form. We can say that the goal of stemming is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. Removing stop words and [5] [6] stem words is done to increase the effectiveness of the system by saving time and disk space. We are using Porter stemming algorithm for stemming purpose. Pre-Processing is done to have optimised matching template of user asked question.

B. Answer Discovering Module

1. 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. 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. 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.

2. Web-Data Extraction Module

The system is developed in such way that if user asked question is out of close domain then search engine will search for appropriate answer and then system will return the accurate answer to the end user. Many sites now support APIs that enable computer programs to harvest information. Several Web-Scrapping solutions are available. For web data extraction we are using JSON which translate HTML into other format and makes it simpler to extract the desired content. JSON is a universal, language-independent format for data. It is based on object-literal notation of JavaScript.

C. Answering Module

Since each and every template representing a question are restored in a database with its answer, just when the best matched template for the question is found, [2]the corresponding answer will be returned to the end user via SMS.

III. TEMPLATE MATCHING

As mentioned earlier, the user questions are answered using template matching. In this section, we discuss the templates used and their syntax. [1]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.

[1]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. 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. [1]In Table I, we tabulate the syntax used for the templates of the questions.

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.
$	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.[1]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. [3]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 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. Instead of linear search, to reduce the matching count between the user asked query and database stored template, we have developed following algorithm

Pre-requisites:

1. After completion of Pre-processing stage we have keywords.
2. [7]We have calculated term frequency of each word stored in the database.
3. Depending upon term frequency, we have constructed two clusters of higher priority words and lower priority words.

Input:

Set of pre-processed keywords.

Algorithm:

1. Determine the priority of pre-processed keywords.
2. Higher Priority keywords stored in tempHigh_str_arr[] and lower priority keyword are stored in tempLow_str_arr[].
3. Search matching template from the database on the basis of tempHigh_str_arr[].
4. From the result set, search for lower priority keywords from tempLow_str_arr[] in template for Best match.
    a. If lower priority keywords found, retrieve the answer mapped to that template.
    b. If lower priority keywords not found, depending upon matched slots i.e. count of matched keywords, template having higher matching slots will be efficient template.

IV. TECHNOLOGIES USED IN SYSTEM

A. Now Sms Lite

The Now SMS/MMS Gateway (NowSMS) is an SMS and MMS content delivery solution. NowSMS is a fast track to

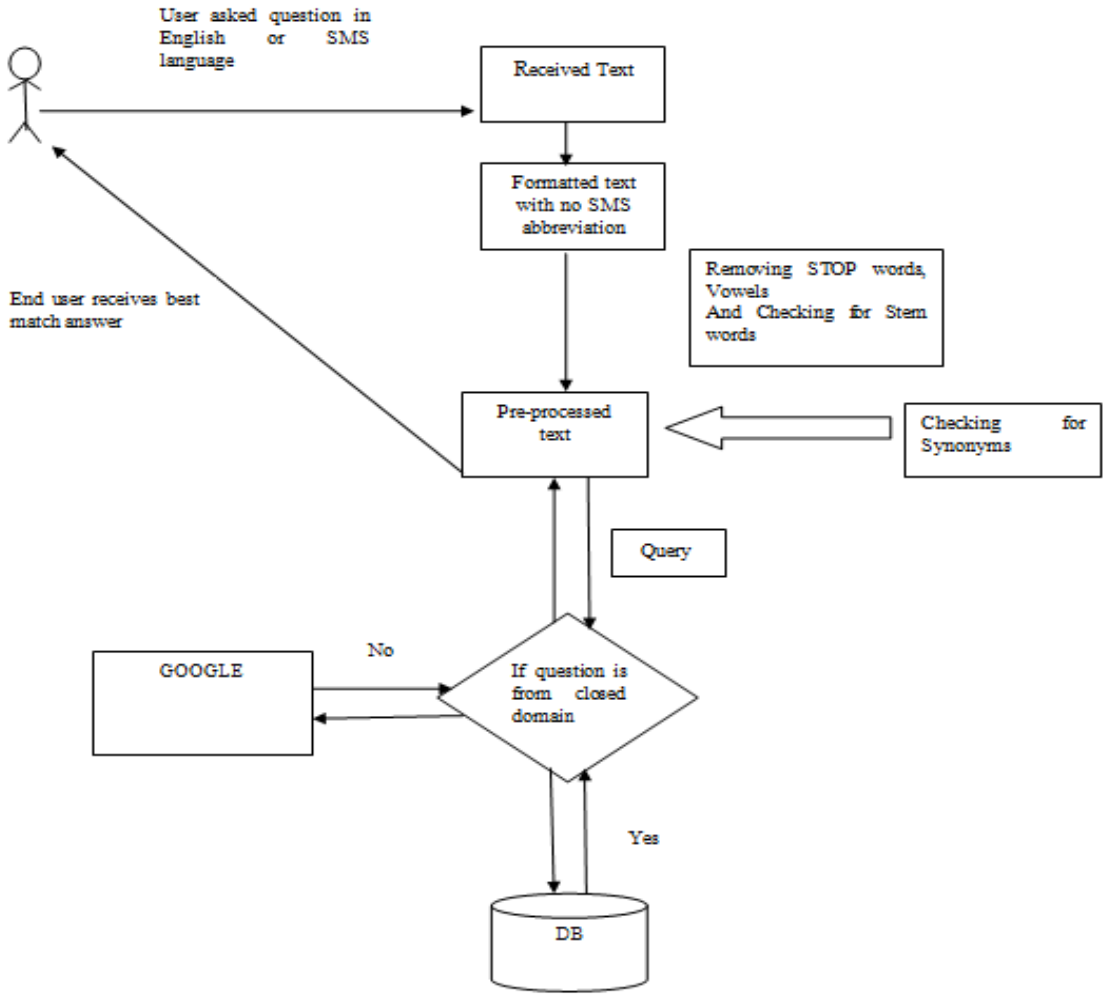

Fig.1.System architecture of the question answering system

deploying and developing SMS, MMS and WAP Push solutions. The Now SMS Lite edition is designed to send and receive SMS and MMS messages using a single GSM (GRPS/EDGE/3G) modem. The full version of NowSMS supports multiple modems, and additional SMSC and MMSC protocol connectivity including support for SMPP, UCP/EMI, CIMD2, and HTTP SMSC connections, plus MM1, EAIF, MM4 and MM7 MMSC connections. The full version of NowSMS is also a fully functioning MMSC. The NowSMS Lite Edition allows clients to submit SMS messages to NowSMS for delivery via the GSM modem, using either the HTTP or SMPP protocols. This document also provides examples for submitting SMS messages to NowSMS from Java, PHP and from a command line interface. Received SMS messages can be routed from NowSMS to an application program using either HTTP, SMPP, or a command-line interface.

V. CONCLUSION

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.

REFERENCES

[1] Sampath Deegalla and Roshan Ragel, "An automatic answering system with template matching", University of peradeniya 978-1-4244-8551-2/10/$26.00 ©2010 IEEE20400, 2010.

[2] 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.

[3] 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

[4] Jaiwei Han, Michline Kamber and Jian Pei, "Data Mining" ,Morgan Kaufmann Publications 3rd edition , 2007

[5] Porter, M.F. (2001), Snowball: A Language for Stemming Algorithms. Available at: http://www.snowball.tartarus.org/texts/introduction.html

[6] sWillett, P. (2006) The Porter stemming algorithm: then and now. Program: electronic library and information systems, 40 (3). pp. 219-223.

[7] Juan Ramos,"Using TF-IDF to Determine Word Relevance in Document Queries",Department of Computer Science, Rutgers University, 23515 BPO Way, Piscataway, NJ, 08855