Performance Evaluation Techniques for an Automatic Question Answering System

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

***Abstract*—Automatic question answering (QA) is an interesting and challenging problem. Generally such problems are handled under two categories: open domain problems and close domain problems. Here the challenge is to understand the natural language question so that the solution could be matched to the respective answer in the database. In this paper we used a template matching technique to perform this matching. The first part of the paper discusses about an automatic question answering system that we have developed using template matching techniques. The approach adopted is an automated FAQ (Frequently Asked Question) answering system that provides pre-stored answers to user questions asked in ordinary English and SMS language. The system also has techniques to overcome spelling and grammar mistakes introduced in questions by its users and therefore user-friendly compared to restricted syntax based approaches. The second part of the paper studies three techniques for performance evaluation in the above system which are based on template matching approach: 1) Random classification of templates, 2) Similarity based classification of templates, 3) Weighting template words.**

***Index Terms*—Evaluation technique, question answering, NLP, template matching, FAQ, answering system.**

I. Introduction

Developing mechanisms for computers to answer natural language questions is becoming an interesting and challenging problem. Such mechanisms allow users to ask questions in a natural language and give a concise and accurate answer. A QA system normally is a computer program, which queries a structured database or an unstructured dataset to get the correct answer. Most QA systems rely on complex architectures including mining text portions and search in textual databases. Therefore, the answer for a question can be found in any resource such as a particular text document, a collection of documents, a collection of web pages, or a knowledge base of information.

The approach we have adopted in this project is an automated FAQ 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 and matching synonyms.

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.
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, these systems usually have much more data available in the public domain from which to extract the answer.

As depicted in Fig. 1, there exist two methods [1], [2] for querying the answer for user questions. Those are, AI method and FAQ search method.

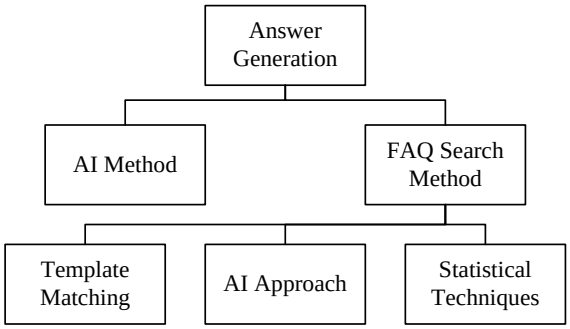

Fig. 1. Different techniques for obtaining answers.

AI method [2]: This requires complex and advanced linguistic analysis programs. This method focuses on answer generation by analyzing questions and creating an understanding of the question

FAQ Search Method [3]: There are three generic methods that an answer can be generated using stored FAQs search method:

1) Artificial intelligence approach This method uses an ontology-based knowledge base
2) Statistical techniques this method considers 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.
3) Template matching

Our System uses a closed domain, FAQ search method coupled with the template matching technique. With the increase of Information and Communications Technology, mobile phones have become a fast and convenient way to communicate over a network. In our system, questions can be

Manuscript received October 22, 2014; revised February 25, 2015.

T. Gunawardena is with the Department of Mathematics and Computer Science, University of Basilicata, Italy (e-mail: etilani@gmail.com).

N. Pathirana is with the University Politehnica of Bucharest, Romania (e-mail: nishara.pdn@gmail.com).

M. Lokuhetti is with the IFS Software Company, Sri Lanka (e-mail: medhavimpl@gmail.com).

R. Ragel and S. Deegalla are with the Faculty of Engineering, University of Peradeniya, Sri Lanka (e-mail: roshanr@pdn.ac.lk, dsdeegalla@pdn.ac.lk).

asked as short messages using SMS (Short Message Service) [4]. Through this extension we enable end users to access information regardless of their location and time, which is more convenient to them.

From the user's perspective the problem is to find the best suitable answer from any resource for a particular question. Ideally we need to measure the answers in terms of being correct and concise. Therefore performance evaluation has been recognized as a particularly important issue for automatic answering systems. In this paper we mainly discuss about performance evaluation. Here, we are reporting on some experimental semi- automated evaluation techniques for QA systems based on answering with a template matching technique. For this case the question and answer pair should be stored in the database for frequently asked questions.

II. Related Work

Automated Question Answering [5] discusses template-based approach in details. Text Retrieval Conference (TREC) [6]. Its purpose was to support research within the information retrieval community by providing the infrastructure necessary for large-scale evaluation of text retrieval methodologies.

Maybury [7] has discussed the characteristics of QA systems and resources needed to develop and evaluate such systems. Although, most QA systems are based on Web environments, SMS has also been used as an environment in contexts such as in learning [8] and agriculture [9].

A well-known template-based natural-language question-answering system is Ask Jeeves (http://www.ask.com).

Many studies have been carried out in computer science on performance evaluation of Question Answering Systems in three ways:

1) **Manual evaluations**-Involve a large amount of human effort and therefore costly since every answer has to be judged by human experts.
2) **Semi-automatic evaluations**-(Breck et al., 2000) [10] presents a semi-automatic approach which relies on computing the overlap between the system response to a question and the stemmed content words of an answer key. Answer keys are manually constructed by human annotators using the Text Retrieval Conference (TREC) question corpus and external resources like the Web. From this method they have showed that their automatic evaluation agrees with the human 93%-95% of time.
3) **Fully automatic evaluations** (Leidner et al) propose [11] a new fully automatic evaluation technique: that is, the intrinsic knowledge in a question-answer pair is used to measure the performance of question answering systems without having to resort to human-created answer keys.

(Magnini et al., 2002) uses a combination of strict and lenient validation patterns against the Web as an oracle. A lenient pattern is meant to retrieve answer candidates, quite like in the QA system itself, whereas the strict pattern is meant to measure the degree of justification via the number of hits.

According to the TREC evaluation on automating evaluation for QA systems the method was to use a set of hand designed "Answer Patterns" in which for every question, five answer candidates have been examined. And also a set of regular expressions that describe answer patterns was defined to automate the evaluation process. To do this an information retrieval system could be used by a human expert to find answers and he would then refine the regular expression pattern in order to include all the answers contained in the collection.

Bilotti *et al*. [12] presents a methodology for scenario QA evaluation including a methodology for building reusable test collections for scenario QA and metrics for evaluating system performance over such test collections.

III. Motivation

A number of question and answering systems have been developed using a variety of techniques. We have developed An Automatic Answering System [13] with Template matching for Natural Language Questions. Therefore, the motivation behind this was to find an effective evaluation technique to identify the most accurate method for question answering while evolving the system.

IV. System Architecture

As in Fig. 2 and Fig. 3 the system architecture of our QA system consists of three main modules,[I] Pre-processing, [II] Template matching and [III] Answering Module.

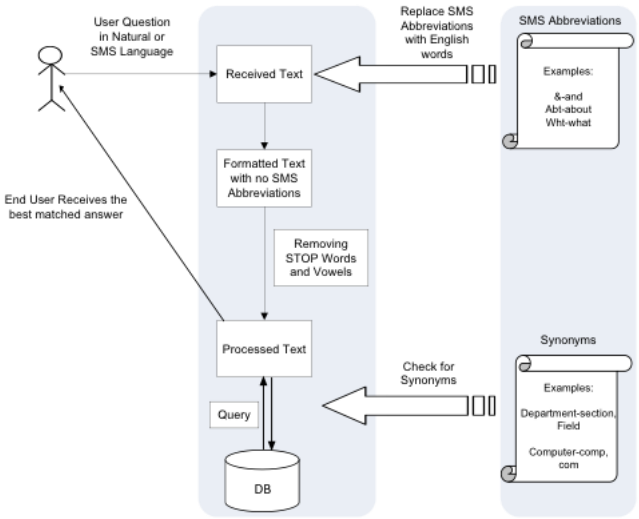

Fig. 2. System architecture.

Pre-Processing Module: Pre-processing module include mainly three operations

1) **Converting SMS abbreviations into general English words:** 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.
2) **Removing stop words:** Stop words are the words that have 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. So considering closed domain we have to consider these words according to our domain.

3) **Removing vowels (disemvoweling):** Disemvoweling [14] is a common feature of SMS language. The purpose of removing vowels is to make it easier to handle spelling mistakes.

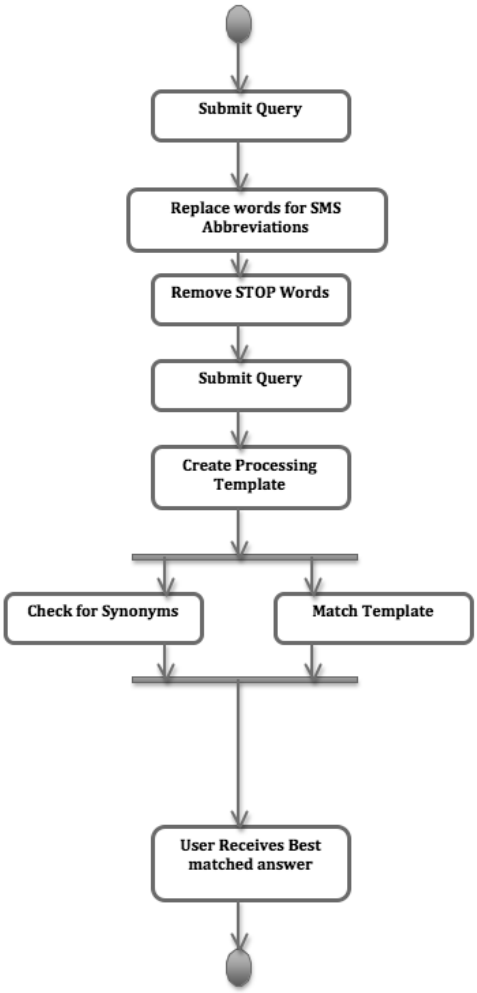

Fig. 3. System architecture.

Template Matching Module: In the database templates are created according to the syntax described under section IV, when the specific pre-processed question/text is queried in the database it is matched against each and every pre stored template until it finds the best-matched template with the received text. 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 [15]. 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 asked by user.

Answering Module: This returns the answer of the best-matched template.

A. Syntax for Template

The success of the question answering thus depends on the quality of templates. The main idea of a template is to match many different variants of a question to a single template. A question might be asked in different ways due to one or more of the following reasons:

1) Different tenses
2) Singular/plural forms
3) Synonyms
4) Order of words
5) Usage of optional words

The syntax used for the templates of the questions are tabulated in Table I. 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. Advantages of using a template matching approach are, (1) precision is high because the keywords are selected using human intelligence and (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.

The main disadvantage of the system is that the templates need to be written manually for all 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.
' '	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.

B. How to Create Templates

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]

The above example illustrates how a template is evolved to a point that it can handle many variants of a single question. Therefore according to above example the final template that goes to the database is *whr/[hw#(g gt)];[cmptr:$dprtmnt*], after removing vowels from the template (*where/[how#(go get)]; [computer:$department]).* Even though this example covers only five questions, in reality it can be the solution to many forms of the same question. A single template that matches many questions can be formed easily using the described syntax in Table I. However this requires a basic understanding of the domain.

C. How to Enhance Template Matching

The template matching technique is enhanced using two additional techniques and they are, (1) 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 in a sentence.

It is important to list synonyms for each term since users might often query using different terminologies. 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:

1) $department-department*, section*, building*, room*, lab*, field

   Synonym file:

   dprtmnt-dprtmnt*, setn*, bldng*, rm*, lb*, fld*
2) $go - get, reach, find
3) $describe - describ*, depict*, illustr*, specif*, character*, clarif*

To improve the quality of the synonyms list, we also have identified the usage of WordNet [5] through which we can expand our query terms.

D. How System Works-Example

We have deployed and tested our system real-time in an Engineering Exhibition Evaluation [16], 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 [16] ran for seven days and the system was used by thousands of visitors.

Table II contains several question covered from one specific template. Not only the question mentioned above but also there are many other questions that can ask from the sample template:

1) Final template inserted to the database after disemvoweling is,

   whr/[hw#$g]/lctn;[cmptr;$dprtmnt]
2) Extraction from the SMS abbreviation file

   Example:

   dep-department
   dept-department
   deptmnt-department
   xbtn-exhibition
   xbtion-exhibition
   xibitn-exhibition
   xbition-exhibition
   xbts-exhibits
3) Extraction from the Synonym file

   Example:

   g – gt, rch, fnd
   ($go - get, reach, find)
   dprtmnt-dprtmnt, sctn*, bldng*, rm*, lb*, fld*($department-department*, section*, building*, room*, lab*, field*).*

TABLE II: SAMPLE TEMPLATE AND MATCHED USER QUESTIONS II

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

V. EVALUATION

In this section we discuss a number of techniques we developed to test the system for getting the correct answer to a question. We have used the questions received from testing system in the Engineering Exhibition [16]. There are many techniques to do evaluation in QA systems [1], [7], [10], [12], [17]. Formally, an FAQ comprises of individual question answer pairs. For answering systems based on template matching, we are discussing three main techniques, (1) Random classification templates, (2) Similarity based classification of templates and (3) Weighting template words.

A. Random Classification of Templates

In Radom classification of templates approach, the templates to be in database at a considered time is chosen randomly. Fig. 4 describes how the templates are randomly classified and stored in the database and results are stored in Table IV.

B. Similarity Based Classification

For this method, templates are clustered based on similarity. That is all the templates in a particular cluster relate to each other in some manner. In this method each cluster group can contain different number of templates. So the total number of templates to be in the database in a particular instance of time are selected based on the templates ratio of clusters.

C. Weighting Template Classification

When considering some user asked questions some word in the sentence has more potential to get the required answer. For example in the question *where is the computer department;*"computer department" has much potential to getting the required answer whereas "where/how" do not have much potential. In the weighting words method all the words in templates are weighted and stored in the database. Here each keyword in a template is weighted considering

their importance to a particular question. When the system receives a question the total weight is calculated. Following example shows the template for a question of asking for the location of computer department.

where/[how#$go]/location(1);[computer:department](2)

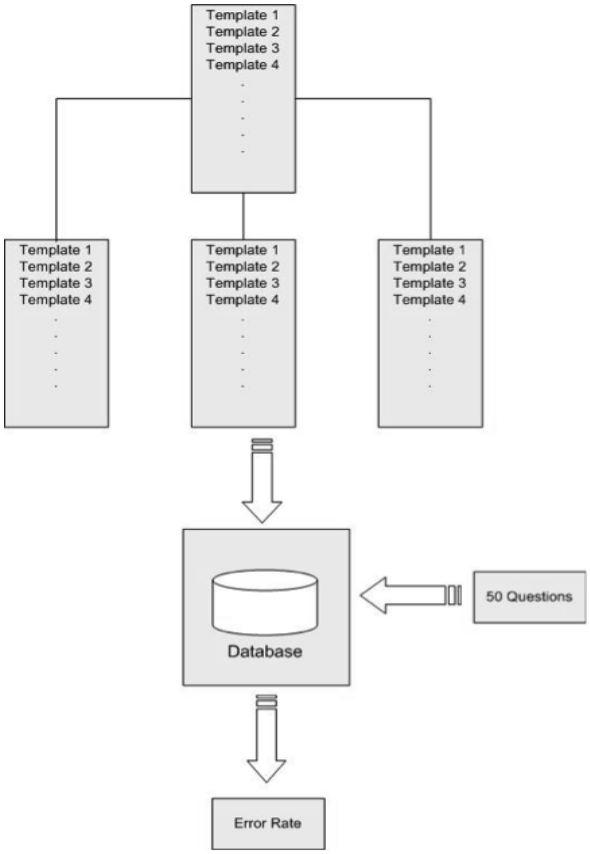

Fig. 4. Random classification of templates.

TABLE III: 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**

VI. RESULTS

As a case study we tested the Question and Answering with template matching system we developed in the Engineering exhibition [Engex2010]. Results are tabulated as in the Table III.

As we can see in Table III, if there is no 100% match for a specific question, the next best solution is returned as the answer. We were able to get 91% accuracy for 100 questions we tested. 100% accuracy can be achieved by changing the templates for the incorrect answers. So the next time can achieve 100% correctness. The accuracy increases with the number of templates stored in the database. We evaluated the system using above evaluation methods. For this case we used a sample with fifty questions with known answers and for the database we used 150 templates.

A. Random Classification of Templates

For randomly classified templates, the templates are selected randomly from the database while changing the number of templates at a time. For every instance of the database we sent the fifty questions with known answers to the system and measured the accuracy rate. The resulted curve is shown in Fig. 5 and the data set shown in Table IV.

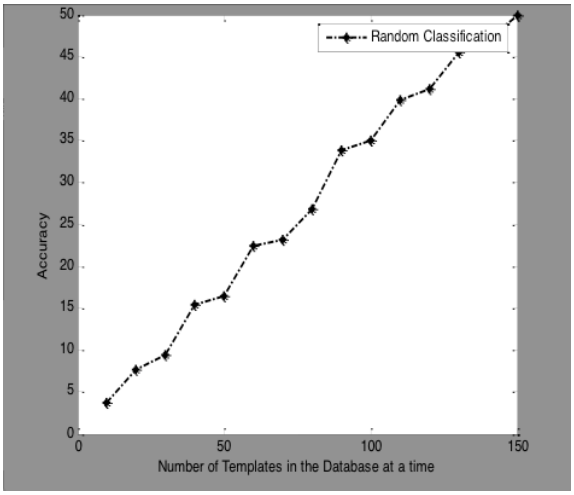

Fig. 5. Accuracy of FAQs for random classification; 50 questions.

TABLE IV: TESTED RESULTS FOR DIFFERENT NUMBER OF TEMPLATES IN THE DATABASE AT A TIME-RANDOM CLASSIFICATION OF TEMPLATES (TESTED QUESTIONS = 50)

#Templates in Database	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150
# of Time															
1	12	7	11	11	18	21	20	37	38	37	36	43	42	49	50
2	2	2	10	27	14	25	21	33	36	28	45	40	45	43	50
3	0	17	3	10	21	17	35	22	23	32	42	31	46	48	50
4	3	4	22	7	17	32	24	26	38	36	39	38	47	46	50
5	9	5	5	18	15	15	14	14	33	32	26	44	47	48	50
6	0	9	6	12	14	26	31	34	36	37	45	47	45	47	50
7	2	8	6	19	28	33	21	27	29	36	38	42	47	48	50
8	3	11	4	27	7	23	26	26	31	32	42	39	48	49	50
9	2	5	15	18	17	13	24	23	36	42	39	46	41	46	50
10	4	9	12	5	14	20	15	27	39	38	46	42	48	47	50
Total/10	3.7	7.7	9.4	15.4	16.5	22.5	23.1	26.9	33.9	35	39.8	41.2	45.6	47.1	50
Percentage (%)	**7.4**	**15.4**	**18.8**	**30.8**	**33**	**45**	**46.2**	**53.8**	**67.8**	**70**	**79.6**	**82.4**	**91.2**	**94.2**	**100**

B. Similarity Based Classification

For similarity based clustering, previous fifty questions were used. But this time, the templates were selected considering their similarity. Therefore the templates those were likely to go into the same group were put together. Likewise we had about 10 clusters for 150 templates. Each cluster contains different number of templates. Therefore we randomly select templates in each cluster according to ratio of total templates of cluster. The results are shown in Fig. 6.

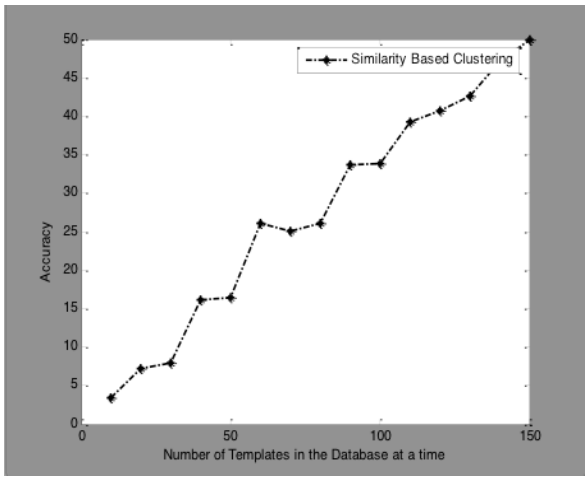

Fig. 6. Accuracy of FAQs for random classification; 50 questions.

C. Weighting Words in Templates

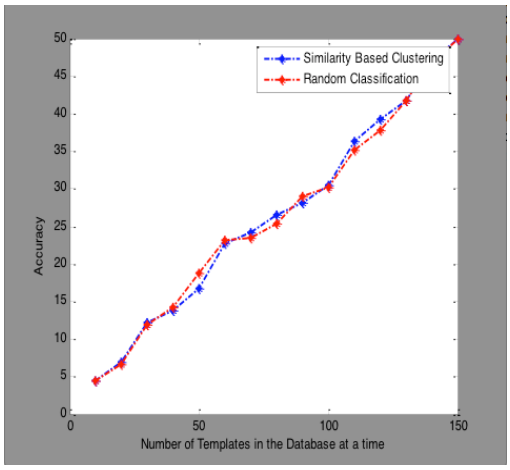

Fig. 7. Accuracy of FAQs for random and similarity based classification with weighted templates; 50 questions.

For this method, same fifty questions used with known answers. In this method all the questions were tested using, 1) Random classification and 2) Similarity based classification. The results are shown in Fig. 7.

VII. Advantages and Comparisons between the Methods

According to the test results, clustering templates based on similarity approach shows the highest accuracy but that is when each cluster having the same number of templates. In our system if there is any difference between the accuracy of each method, it is because several questions can rely on one template. Therefore if the template is not in the database the system fails to reply with the correct answer. Hence this random classification of templates could be less accurate than similarity based classification.

Weighting method and the other two methods are two separate techniques. Random classification and similarity-based classification were tested with and without using weighted templates. Therefore it is not easy to compare the three approaches at once. It is possible to get a higher accuracy using weighted template with clustering method.

VIII. Conclusion and Future Work

Our experiment shows the three different techniques we can use when evaluating answering systems. Similarity based clustering method and random classification method can be used individually or with the weighting template method. When the database has more templates it gives better accuracy. In practical situations where a large number of templates are used, similarity based clustering method with weighed templates would be the better choice according to the experiment results. The accuracy for this method can be improved by having similar number of templates for each cluster and by enhancing the template weighting mechanism.

As future work we can enhance the weighting method with different techniques and also generate templates automatically.

Acknowledgment

The authors would like to thank all the academic staff, non-academic staff of faculty of Engineering, University of Peradeniya for their generous support and Department of Computer Engineering for providing financial support for the project.

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*, pp. 84-85, December 2008.

[2] J. Palme. (2008). Principles of Intelligent Natural Language Question Answering. [Online]. Available: http://web4health.info/en/answers/project-search.htm

[3] J. Palme and E. Sneiders, *Natural Language Question Answering System Classification Manual*, Aug. 2003.

[4] SMS language. (2013). [Online]. Available: http://en.wikipedia.org/wiki/SMS_language

[5] A. Andrenucci and E. Sneiders, "Automated question answering: review of the main approaches," in *proc. the 3rd International Conference on Information Technology and Applications (ICITA'05)*, vol. 1, pp. 514-519, July 2005.

[6] Text retrieval conference. (2012). [Online]. Available: http://trec.nist.gov

[7] M. T. Maybury, *Toward a Question Answering Roadmap*, 2002.

[8] S. Balasundaram and B. Ramadoss, "Sms for question-answering in the m-learning scenario," *Journal of Computer Science* 3, vol. 2, pp. 119-121, 2007.

[9] M. Suktarachan, P. Rattnamanee, and A. Kawtrakul, "The development of a question-answering services system for the farmer through sms: query analysis," *in Proc. the 2009 Workshop on Knowledge and Reasoning for Answering Questions*, Suntec, Singapore, Aug. 2009, pp. 3-10.

[10] E. J. Breck, J. D. Burger, L. Ferro, L. Hirschman, D. House, M. Light, and I. Mani, "How to evaluate your question answering system every day and still get real work done," The MITRE Corporation, 2000.

[11] J. L. Leidner and C. C. Burch, "Evaluating question answering systems using faq answer injection," in *Proc. the Sixth Computational Linguistics Research Colloquium (CLUK-6)*, Edinburgh, UK, 2003, pp. 57-62.
[12] M. W. Bilotti and E. Nyberg, "Evaluation for scenario question answering systems," in *Proc. the International Conference on Language Resources and Evaluation (LREC 2006)*, 2006.
[13] T. Gunawardena, M. Lokuhetti, N. Pathirana, R. Ragel, and S. Deegalla, "An automatic answering system with template matching for natural language questions," *in Proc. 2015 5th International Conference on Information and Automation for Sustainability (ICIAFs)*, Dec. 2010, pp. 353-358.
[14] Disemvoweling. (2012). [Online]. Available: http://en.wikipedia.org/wiki/Disemvoweling
[15] C. Fellbaum, *Word Net, an Electronic Lexical Database*, MIT Press, 1998.
[16] Engineering exhibition. (2010). [Online]. Available: http://www.pdn.ac.lk/eng/engex2010/
[17] L. Hirschmanand and R. Gaizauskas, "Natural language question answering: the view from here," *Natural Language Engineering*, vol. 7, New York, NY, USA, Dec. 2001, pp. 275-300.

Tilani Gunawardena is a final year PhD student at Department of Mathematics and Computer Science in University of Basilicata, Italy. She earned her BSc. engineering (Hons) degree specialized in computer engineering from University of Peradeniya, Sri Lanka. She worked as an instructor in the Department of Computer Engineering, University of Peradeniya, Sri Lanka for one year. Her research interests are GPU computing, cloud computing, big data analysis, neural networks, artificial intelligence, and data mining and machine learning.

Nishara Pathirana is a second year master student in erasmus mundus data mining and knowledge management in the Politechnica University Bucharest, Romania. She received her BSc. Engineering (Hons) degree specialized in the field of computer engineering from university of Per- adeniya, Sri Lanka.

She has worked as a software engineer in DirectFN Technologies (pvt) Ltd, Colombo, Sri Lanka for one and half years and has worked as an instructor in the Department of Computer Engineering, University of Peradeniya, Sri Lanka for about two years. Her scientific interests include data mining, machine learning, artificial intelligence and neural networks.

Medhavi Lokuhetti is a senior software engineer at IFS (Pvt), Sri Lanka. She completed her BSc. engineering (Hons) degree specialized in computer engineering from University of Peradeniya, Sri Lanka. She is currently following an MBA degree in University of Colombo, Sri Lanka. Her research interests are machine learning, data mining and artificial intelligence.

Roshan Ragel is a senior lecturer at the Department of Computer Engineering, Faculty of Engi- neering, University of Peradeniya, Sri Lanka. He received his BSc engineering (Hons) from University of Peradeniya in 2001 and his PhD in computer science and engineering (UNSW, Jun. 2007). His research interests are micro-architectural aspects of embedded systems design and their security and reliability issues.

Sampath Deegalla is a senior lecturer at the Department of Computer Engineering, Faculty of Engineering, University of Peradeniya, Sri Lanka. He received his BSc engineering (Hons) from University of Peradeniya and his Ph.Lic from Stockholm University, Sweeden. His research interests are machine learning and data mining issues.