An Improved Answer Retrieval System Taping the Linkage Structure for Noisy SMS Queries

Gaurav Batra
Department of Computer Science
College of Engineering Roorkee
Roorkee, India

Mansi Goel
Department of Computer Science
College of Engineering Roorkee
Roorkee, India

ABSTRACT
The advent of technology has not only tailored the scientific and research work but has also sufficed to the needs of a commoner. Although communication and online support has become quite contemporary yet it requires huge amount of human resource to meet the varied demands. In this paper a server application is proposed which provides automated support for the customer queries via Short Messaging Service (SMS). It facilitates a layman with an instant answer to any of his query. The system is highly capable of handling inherent noise present in the queries and also taps their syntactic and semantic structure. This approach performs two functions simultaneously. One of which is handling of noise and inferring the best possible question, which the user actually meant to ask. Second is to match the refined question with the existing database of questions and then provide a corresponding answer.

General Terms
Algorithm, Design, Experimentation, Performance.

Keywords
FAQ, Noise, Query Extraction, Question Answering System, Retrieval.

1. INTRODUCTION
For a long time, technology had confined itself to the clutches of sophisticated people. Common man was rather kept aloof from it. Gradually researchers realized to acquaint their research work to the commoner. One such effort came in the form of mobile phone. Since its invention, mobile phones have gained tremendous popularity in every strata of society. And With the rising popularity of mobile phones, text messaging has become an integral part of it. There are several reasons behind this worldwide acceptance. One is that Short Messaging Service (SMS) is a quick and reliable means of sending information in text format. Another reason is the language synonymous with it. The massive use of abbreviations makes it one of the easier and faster methods of communication.

Time has never ceased to exhibit its grandness. It is the most precious belonging. It is the time which is replacing man with machines. Due to the inaccuracy and longevity of manual work, machines are substituting humans. Automation is one such example. The use of control systems is to reduce the human effort. Automation has a renowned impact in industries and enterprises. This impact is contributed to the fact that automation substitutes human operators in performing the undertakings that are beyond human potential. It has various applications and one of them is question answering systems.

The system exploits both the above elevated technologies. It is a server application which automates the answer retrieval task via SMS. The retrieval process is such that the user is provided with the five topmost relevant answers to his/her query instantly. The communication starts via SMS from the user. This SMS is received by the system which is installed on central server. It then processes the incoming message. Answer retrieval process deals with two major tasks. In the preliminary step, processing of the message is done in order to eliminate inherent noise. This noise is due to the presence of abbreviations, transliterations, etc. Subsequently, a noise free query is incurred. The refined query is then parsed using the natural language parser providing us with the *Question focus*. The Question focus derived is used to rank the corpus questions, thereby resulting in the precise answer [1].

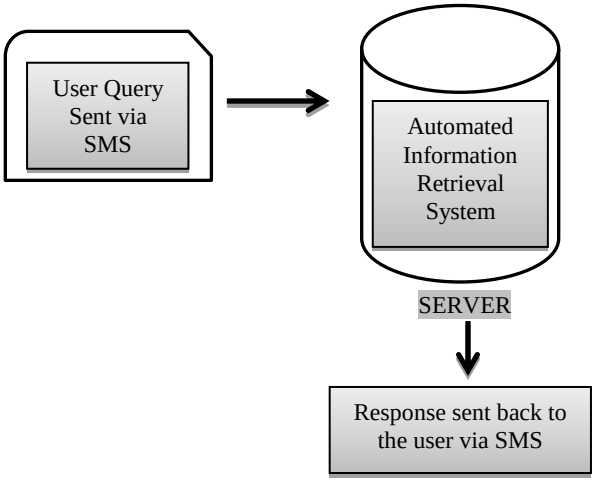

Figure 1: Working of an Automated Information Retrieval System

For generating the best possible response for the user query, two major tasks are to be handled viz. Handling Noise in SMS properly to generate a noise free query & extracting the best matching question from the Dataset comprising of question-answer pairs already available with the server.

Now-a-days use of abbreviations and shorthand writing is common in SMS texting. It saves both time and space in texting. Human is intelligent enough to extract the contextual meaning out of these ambiguous words. However, in order to process these words using a machine, we need to remove the inherent noise.

The accuracy of the query obtained after the removal of noise directly affects the further functioning of the system. Hence, the

more accurate the query will be, the more precisely a response can be generated. Along with the accuracy, time factor has to be taken into account. As the response of the query is to be sent back to the user via SMS, the whole process needs to be finished quickly in order to generate instantaneous response.

Figure-1 shows an automated system installed on the server, which accepts a SMS query sent from a user and this system process it to provide a response back to the user.

2. PRIOR WORK

Significant amount of work has been done in the past to handle inherent noise in the queries. Different attempts in distinct contexts have been made to remove noise from the words. For this basically one of the two mentioned approaches are employed: human expertise based and automated Information retrieval based. Both the approaches have their own advantages and drawbacks. Each one has a specific application. Systems that require human intervention are engaged in the applications where the response time is a major factor and the needs of the user changes considerably. But such a system needs massive resource utilization. Moreover whole information cannot be centralized. In contrast to this, an automated information retrieval system doesn't require any human interaction and generates a noise free text without considering any contextual meaning. So, this type of system finds its utility in the applications where server is provided with an enormous dataset. Also, the updations are not too frequent.

Langer & Banga [2] used Syntactic Tree Similarity approach to handle noisy queries and explained how user satisfaction and response time defines the efficiency of a retrieval system. But the above mentioned paper is not configured to handle all kind of noise like transliterations used in the queries. For example, "someone" is generally written as "som1" in SMS language.

Kothari & Negi [3] adopted a different approach to retrieve the correct response for a query and explained it as a combinational search problem to handle the lexical and semantic noise. Kothari ranked every keyword using Inverse domain frequency (IDF) which gives more weightage to a more frequent term in the corpus. However, it cannot be used as a reliable measure to rank the keywords. There are various cases which support this argument.

Authors in [15] adopted a statistical approach to cater the task of noise removal and question matching. A weighted measure is considered for the extraction of the best variant of the noisy text. A similar approach has been presented in [16] which provide an unsupervised method for extracting domain specific lexical variants for the noisy text. Also the approach proposed is demonstrated on the dataset obtained from twitter to remove noise from the text.

Chen & Subramanian [4] proposed a system named *SMSFind* which is an automated FAQ retrieval system to extract a concise answer after the noise from the query is refined. SMSFind uses a *hint* to determine the context of the query, this *hint* is either provided by the user or is derived from the query itself. But SMSFind server takes more time in generating a response as the query is sent to general search engine following which the related pages are downloaded and answer is extracted from the downloaded pages.

Burke & Hammond [5] explained an automated system named *FAQ Finder* which extracts the best matching FAQ from the database files at server. It also uses synonym lookup using WordNet thesaurus to improve its results. But this system is not configured to handle any kind of noise present in the query and pre-assumes that it is already refined.

Existing approaches have one or the other limitation, and thus our system exploits an approach that employs both the tasks of noise handling and semantic structure recognition.

3. APPROACH AND ALGORITHM

To retrieve the best possible answer for the query, it first needs to be free from any kind of noise. Hence, the system begins with the transposition of ambiguous words with the best matching variants available. Similar approaches have been used in the past for this purpose. However this paper implements those ideas with slight modification. A short description is the following-

- Each word in the query is treated as a separate token.
- A Standard English Language word list is referred for token matching.
- In order to ascertain the best possible variant out of the few lakh words available in the wordlist, 3 algorithms are used viz. Soundex, Longest Common Subsequence and Levenshtein Distance Algorithms.
- *Soundex:* Soundex Algorithm is a measure which categorizes a word on the basis of its phonetic behavior. There are distinct versions of this algorithm available [6] [7].
- *LCS (Longest Common Subsequence):* This algorithm determines the longest common subsequence between the two given strings. It returns an integer indicating the LCS for aforesaid strings [8].
- *Levenshtein Distance (Edit Distance):* Levenshtein Distance algorithm measures the difference among two strings, also known as Edit Distance. This algorithm returns an integer indicating the minimum number of edits required to transform one string into another [9].
- To improve computation time and to achieve high performance, Soundex is applied at the first step**.** This results in a reduced word list. Thereafter, similarity measure is calculated using the LCS and Levenshtein Distance algorithms.

Finally a noise free query is obtained as an outcome (as shown in figure- 2). To determine the best possible match for user question, a corpus (database) of questions is loaded into memory having question-answer pairs constituting different domains.

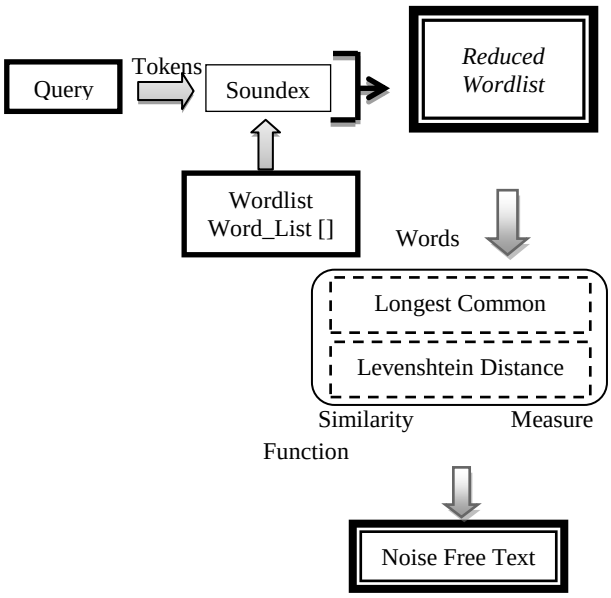

Figure 2: Handling Noise in the query

- The refined query is treated as the final query for which best possible response is to be generated.

3.1 Question Focus

The focal point of any query can be recognized by finding its parts of speech. Hence, to determine the best possible query from the corpus (database of questions available) [10]. To find part-of-speech of the query, it is parsed using a Link parser [11]. The parser returns the relevant information (like part-of-speech) from the query and hence helps in extracting those words in the query which are of prime importance.

This parser considers the relationship amongst the words used in query. The parser has a dictionary of about 60,000 word forms. It can also handle unknown words, if any, in the query simply by neglecting them.

For any query, the link parser returns a linkage structure that furnishes us with the syntactic structure of the query like noun, verb, adjective etc. used in the text.

For example, in the sentence - "How to apply for Passport?". For the above question the resultant keywords (*Question focus*) are– "apply" and "passport". This information is saved in a different file for further use.

3.2 Providing Weightage to Keywords

The Keywords extracted from the query are matched with the words present in the corpus and accordingly, a weightage is provided to each word.

- If the word is present in the Dataset (corpus) then it is given highest weightage of 2.

And if the word is not present in the Dataset, then it is searched for its synonyms using a popular thesaurus known as WordNet [12]. WordNet not just considers word forms, strings, but also consider the contextual meaning of the word [13]. WordNet generates a list of synonyms for a word. These synonyms are again matched with the words present in the Corpus.

- If the synonym is matched with some word in the corpus, then the actual word in the query is replaced with this synonym and is given a weightage of 1. The less reliability of the resulting synonym is the reason for the less weightage.

3.3 Extracting Answers:

After applying above approach repeatedly, every keyword has some weightage. Now every question in the corpus is checked for these keywords and a ranking is given to every question on the basis of the weightage defined for every keyword.

For example, consider the keywords and their corresponding weightage: "apply" (2), "visa" (2), "reserve" (1). Let one of the questions in the corpus be "How to apply for visa". Only two keywords are matched with the words in the question. Hence the total weightage for this question comes out to be 2+2=4.

Similarly, all other questions in the database are ranked. And finally the question with the maximum total weightage is selected as the best matching question in the corpus for the query (as shown in figure-3). And finally, the answers corresponding to these best matching questions are sent back to the user as the response to his query [14].

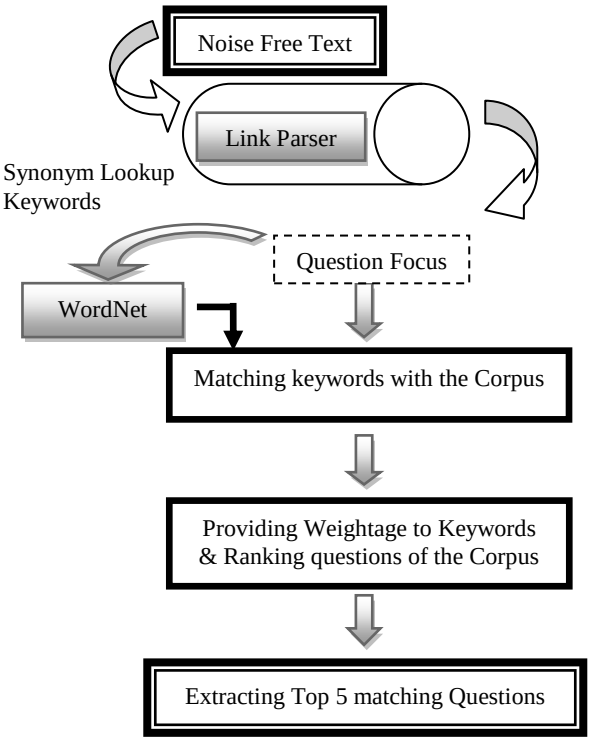

Figure 3: Extracting best matching questions from the corpus

4. RESULTS AND CONCLUSION

The main purpose of the server application described in this paper is to facilitate the common man with a service through which a mobile user can ask his/her queries via SMS. In response, the server facilitates the user with a precise and instant answer using an automated software system. The approach includes two basic stages to achieve the desired results: Removal of noise present in the SMS query sent by the user and Extraction of the most relevant answer from the dataset available to the server.

Table 1: Similarity measure function for the token "srvc"

Word from the wordlist	*LCS_ratio*	*Edit_distance*	*Similarity_measure*
Shropshire	0.285714	9	0.031746
Sharps	0.400000	5	0.080000
Service	**0.800000**	**4**	**0.200000**
Services	0.666667	3	0.133333
Sarapes	0.400000	6	0.666667

Let's consider an example query "Wht typ of srvc s avlbl fr carer conslng". The token "srvc" is checked for the words having the identical Soundex code in the wordlist and a reduced wordlist of 257 words is generated. Now similarity measure only for these words is calculated to find the best variant using the standard LCS ratio and Edit Distance algorithms. Table-1 displays the results of the three algorithms for the token "srvc". The maximum value of the similarity measure comes out for the word "service" in the wordlist which is highlighted. And hence is replaced in the final query obtained. Similarly, "typ" is replaced with "type", "avlbl" with "available" etc. Finally, a noise free query is generated and used for extraction of the response (as shown by figure 4).

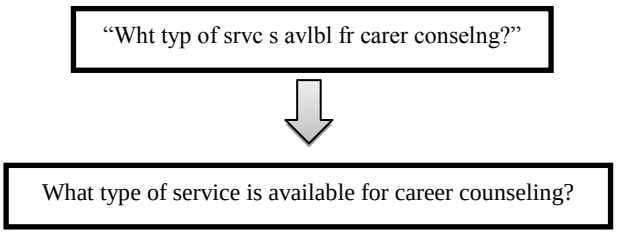

Figure 4: Query before & after noise removal

This refined query is then parsed using Link Parser, which generates a syntactic structure of the sentence providing relevant information to determine the question focus. For example, different structures generated for the above example query are:

[S What type of [NP service NP] [VP is [NP available for career counseling NP] VP]? S]

[S What type [PP of [NP service NP] PP] [VP is [ADJP available [PP for [NP career counseling NP] PP] ADJP] VP] ? S]

Where S is the Subject, NP is the Noun Phrase, VP is the Verb Phrase, PP is Preposition Phrase, and ADJP is adjective Phrase etc.

After Determining the Question Focus, weightage to all the keywords is provided and then questions are ranked accordingly. For better results, top five matched questions from the corpus are listed for the user query in figure-5.

As this is an automated system, there may be cases where there could be possibility of having more than one questions as the best matching response. So, to handle that possibility this system provide user with top 5 responses generated by the application.

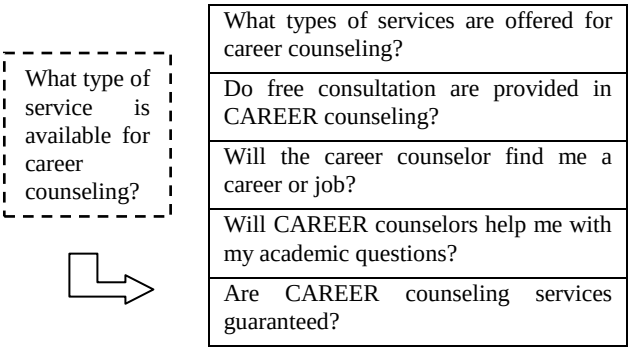

Figure 5: Top five results listed for a query

5. FUTURE WORK

No functioning system can be perfect. There is always a scope of refinement. Our system is no exception. Current work involves handling the noise in SMS queries and finding its linkage structure to provide an accurate answer. However, further enhancements can be done on the information retrieval process. One way is to maintain a database of past queries and links of their respective answers which were answered correctly. It may lead to a further decrease in the response time.

Our system also rules out the case when there may be many words in the query whose synonyms are searched in WordNet. Since there is a list of synonyms of each word. Permuting the different synonyms may generate a large number of questions. Thus leading to a huge amount of time to search a matching question in the available database.

Also the system can be made more versatile by answering multi-lingual or cross-lingual queries. Thus enabling the customer to place the query in other regional languages also.

6. REFERENCES

[1] Danish Contractor, Govind Kothari, Tanveer A. Faruquie, L. Venkata Subramaniam, Sumit Negi. "Handling noisy queries in cross language FAQ retrieval" EMNLP'10 Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing.

[2] Akhil Langer, Rohit Banga , Ankush Mittal, and L.V.Subramaniam . "Variant Search and Syntactic Tree Similarity Based Approach to Retrieve Matching Questions for SMS queries" Proceeding AND '10 Proceedings of the fourth workshop on Analytics for noisy unstructured text data.

[3] Govind Kothari, Sumit Negi, Tanveer A. Faruquie, Venkatesan T. Chakaravarthy, L. Venkata Subramaniam. "SMS based interface for FAQ retrieval" ACL '09 Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2.

[4] J. Chen, L. Subramanian, and E. Brewer, "SMS-Based Web Search for Low-end Mobile Devices," in MobiCom '10. New York, NY, USA: ACM, 2010, pp. 125–136.

[5] Robin D. Burke, Kristian J. Hammond, Vladimir A. Kulyukin, Steven L. Lytinen, Noriko Tomuro, Scott Schoenberg, "Question Answering from Frequently-Asked Question Files: Experiences with the FAQ Finder System" *AI Magazine* Vol. 18 Pg. 57--66

[6] "The Soundex Indexing System" National Archives and Records Administration. 2007-05-30. Retrieved 2010-12-24.

[7] Soundex Agorithm "phonetic-string-comparison-with-soundex" http://www.blackbeltcoder.com/Articles/algorithms/phonetic-string-comparison-with-soundex

[8] Cormen, Thomas H.; Leiserson, Charles E., Rivest, Ronald L., Stein, Clifford (2009) [1990]. Introduction to Algorithms (3rd Ed.). MIT Press and McGraw-Hill. ISBN 0-262-03384-4.

[9] "Levenshtein Distance, in Three Flavors" by Michael Gilleland, http://www.merriampark.com/ld.htm.

[10] Praveen Kumar, Shrikant Kashyap, Ankush Mittal, And Sumit Gupta . International Jl. on E-Learning (2005) 4(1), 149-166.

[11] Temperley, D., Sleator, D., & Lafferty, J. (1993) Parsing English with a link grammar. Third Annual Workshop on Parsing Technologies.

[12] Gonzalo, J., Verdejo, F., Chugur, I., & Cigarran, J. (1998). Indexing with WordNet synsets can improve text retrieval. Proceedings of the COLING/ACL '98 Workshop on Usage of WordNet for NLP.Montreal, Canada, 38-44.

[13] WordNet Structure - http://wordnet.princeton.edu/

[14] Cody Kwok, Oren Etzioni, Daniel S. Weld, "Scaling question answering to the web" ACM Transactions on Information Systems (TOIS) Volume 19 Issue 3, July 2001.

[15] D. Contractor, T.A. Faruquie, and L.V. Subramaniam, 2010. Unsupervised Cleansing of Noisy Text. In Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pages 189–196. Association for Computational Linguistics.

[16] Stephan Gouws; Dirk Hovy; Donald Metzler. "Unsupervised Mining of Lexical Variants from Noisy Text" *Proceedings of the First workshop on Unsupervised Learning in NL*. July, 2011. Association for Computational Linguistics, Page 82-90. http://www.aclweb.org/anthology/W11-2210