Variant Search and Syntactic Tree Similarity Based Approach to Retrieve Matching Questions for SMS queries

Akhil Langer
Electronics and Computer Engineering Department
Indian Institute of Technology
Roorkee, India
akhilanger@gmail.com

Rohit Banga
Electronics and Computer Engineering Department
Indian Institute of Technology
Roorkee, India
iamrohitbanga @gmail.com

Ankush Mittal
Computer Science Department
College of Engineering
Roorkee, India
dr.ankush.mittal @gmail.com

L.V. Subramaniam
IBM India Research Lab
New Delhi, India
lvsubram@in.ibm.com

ABSTRACT

Community based Question Answering archives have emerged as a very useful resource for instant access to comprehensive information in response to user queries. However, its access remains restricted to internet users. Access to this resource through Short Message Service (SMS) requires that a high precision automatic similar question matching system be built in order to decrease the search time by decreasing the number of SMS exchanges required. This paper proposes a solution that handles inherent noise in SMS queries through variant search, modeling the problem as one of combinatorial search. Following this, it uses syntactic tree matching to improve the ranking scheme. We present our analysis of the system and conduct experiments to test its feasibility. Experiments show that our approach outperforms the existing approaches significantly.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval Models, Search Process, Selection Process; H.3.4 [**Systems and Software**]: Question Answering(fact retrieval)systems, Performance Evaluation(efficiency and effectiveness)

General Terms

Algorithms, Design, Experimentation, Performance

Keywords

Question Answering, SMS Queries, Noisy Text, Syntactic Structure, Question Matching, Similarity Scores, Noise Handling

AND'10, October 26, 2010, Toronto, Ontario, Canada.
Copyright 2010 ACM 978-1-4503-0376-7/10/10 ...$10.00.

1. INTRODUCTION

Use of Short Message Service (SMS) for disseminating and gathering relevant information has been widely studied and various applications have been developed that exploit their wide accessibility. Being accessible through all mobile phones and because of its simple interface it can be used by anyone. SMS usage continues to grow every year with more and more new innovative ways of utilizing its potential coming to its forefront. An average Indian sends 29 SMS per month and it is not just being used to keep in touch with friends but also to check bank balances, mobile bills, train schedules, etc. and search features like weather, sports, movies, etc. However these existing SMS based information service providers require the user to be familiar with a fixed format and limited vocabulary to put their queries. This limits the utility of the system.

Automatic handling of the varied forms of user queries not only requires a large database of QA pairs but also the technology to match the user query to the documents in the database. Community Based Question Answering (CQA) has emerged as a very popular resource of QA pairs where people can access historical QA pairs and also put their questions on the site for them to be answered by other users. Within 5 years of its online availability a popular CQA service provided by Yahoo has generated a database of 85M resolved questions. The site reports about 36M unique visitors per month. To enable access to this knowledge accumulated in CQA through SMS requires robust information retrieval (IR) techniques. Three factors that determine how well the IR system works are the effectiveness of results or relevance to the query, satisfaction of the user and processing time (time and space efficiency). In this paper we focus on the first factor i.e. the relevance of the user query to the retrieved query. Particularly, in communication over SMS this factor is very important because in order to enhance user experience the information need of the user should be sought in as few SMS exchanges as possible. The second and the third factor are discussed in Section 7.

In order to improve question search over CQA services, various IR techniques have been developed. Language Models, Language Model enhanced with Question Classification, Syntactic tree similarity, Translation Model, Question Classification, etc. are some of these techniques that have been widely used for similar question retrieval. However, these techniques cannot be directly applied to SMS queries because of the inherent noise present in them in the form of misspellings, non-standard abbreviations, transliterations, phonetic substitutions and omissions. The noise present in the queries makes it difficult to build an automatic question answering system over SMS. On the other hand, restricting the user to type the words correctly on a small screen and keypad affects the user experience and reduces the system usability.

In this paper, we present a solution that addresses the above problems. We present a solution by which the precision of retrieving the appropriate question from CQA dataset for noisy queries can be improved using Syntactic Tree Matching. More specifically, the papers contribution is to show how syntactic tree matching based approach can be applied to noisy SMS queries and thus make feasible the development of a system that allows access to historical QA pairs in CQA archives accessible through SMS.

The organization of the remaining paper is as follows. In Section 2 we discuss the prior work in this domain. In Section 3 a brief overview of the basic retrieval models we have used is given. In Section 4 we present our solution using a stepwise approach. In Section 5 and Section 6 we give the overall retrieval model and the experimental results. In Section 7 we discuss other two important factors of an IR system namely the user satisfaction and response time. We conclude this paper and give a perspective for future work in Section 8.

2. RELATED WORK

As mentioned in Section 1 there has been growing interest in SMS based services. These services have evolved from Human intervention based services to automatic systems that use Information Retrieval and Natural Language Processing techniques.

Google SMS, a suite of SMS based applications includes Google SMS search (news, local weather, sports, agriculture tips, etc.), Google SMS Tips (health tips, clinic finder), Google Trader (”marketplace” application that helps buyers and sellers find each other). Google introduced these services in 2009 in Africa [2] which has the world’s highest mobile growth rate. Other notable services include access to Yellow Page services [5], Email, Blog, etc. In general, these services connect the information seekers to information sources by matching the user query with the description of the content that is associated with the information segment. The level of granularity is high as this information provides information to only Frequently Asked Questions (FAQ) i.e. the questions that are commonly asked. On the other hand, the size of the archives in Community based QA services is so large that it is most likely that our question has already been asked by someone else. Questions varying from ”How to lose weight quickly” to ”How can I make someone lose weight without them knowing” are all present in the archive. Thus, the task is to develop methods to find the best matching question.

There has been a lot of research in similar question retrieval. A host of techniques have been applied to achieve this task. Some of these are lexical and semantic similarity approach [3], translation model [10], vector space model, Okapi, language model. In one of the recent works, Cao et.al.[4] proposed an enhanced language model that uses categorization information for question retrieval from CQA archives. Wang et.al. [11] have proposed a syntactic tree similarity based approach which is semantically smoothed by allowing closely related words to match (using Wordnet Similarity), relaxing the production rules to allow partial matching and using answer matching to bring in more semantically related questions. Their work achieves significant improvement over the simpler Bag-of-Words (BoW) and Tree kernel based approaches.

Noise in search engine queries has also been well studied. Levenshtein Distance augmented by the use of a Language Model (LM) from corpus of web queries, use of web search results to suggest better corrections, etc. are some of the methods that have been proposed for handling noise in search engine queries. But these techniques cannot be applied to SMS queries because of the difference in the type of noise. In online search engines queries the noise can be because of typographical mistakes or misspelled words while in SMS the words are intentionally misspelled for the ease of typing. Therefore, SMS queries require different type of handling. Govind et.al. [6] propose an unsupervised approach for handling noisy lexical and semantic variations in SMS queries. Previous approaches rely on aligned corpus of SMS and conventional language for training which is difficult to build and requires considerable human effort.

3. PRELIMINARIES

Here we cover the retrieval models that we have used.

3.1 Noise Handling in Queries

Levenshtein distance [7] (or Edit distance) is one of the popular techniques used in matching noisy terms in SMS text with the actual terms in the vocabulary. Lucene [1] uses fuzzy searcher based on the Levenshtein distance. However, as reported in [6] with higher value of similarity parameter in Lucene’s fuzzy match, the performance of information retrieval actually degrades. To the best of our knowledge the method proposed by [6] is the latest work in the development of SMS based automatic question answering in which they propose the following similarity measure between a SMS term (s_i) and a term (t) in the domain dictionary:

$$\alpha(t, s_i) = \frac{LCSRatio(t, s_i)}{EditDistance_{SMS}(t, s_i)} \qquad (1)$$

Longest Common Subsequence Ratio (LCSRatio) [8] is the ratio of the length of their Longest Common Subsequence (LCS) and the length of the dictionary term.
$EditDistance_{SMS}$ [9] compares the Consonant skeletons of the dictionary term and the SMS token. A scoring function that determines how closely a question in the corpus matches the SMS string is defined as

$$Score(Q) = \sum_{i=1}^{n}[\max_{t:t\epsilon Qandt\sim s_i} \alpha(t, s_i)] \qquad (2)$$

i.e. For each token s_i, the scoring function chooses the term from Q having the maximum weight; then the weight of the n chosen terms are summed up to get the score. The goal is to find the question Q^* having the maximum score.

3.2 Syntactic Similarity

Purely lexical approaches are often inadequate to perform fine-level textual analysis if the task involves the use of more varying syntactic structures or complex semantic meanings. In order to capture syntactic similarity between the two queries Zhang and Lee [12] proposed a tree kernel method based on the idea of counting the number of tree fragments that are common to parse trees of both queries, and is defined as

$$k(T_1,T_2)=\sum_{n_1 \epsilon N_1}\sum_{n_2 \epsilon N_2} C(n_1,n_2) \qquad (3)$$

Where, N_1 and N_2 are sets of nodes in two syntactic trees T_1 and T_2, and $C(n_1, n_2)$ equals to the number of common fragments rooted in nodes n_1 and n_2. However, to enumerate all possible tree fragments is an intractable problem. The tree fragments are thus implicitly represented, and with dynamic programming, the value of $C(n_1, n_2)$ can be efficiently computed as follows:

$$C(n_1,n_2)=\begin{cases} 0, \text{ if } n_1 \neq n_2, \\ 1, \text{ if } n_1 = n_2 \text{ \& they are terminal nodes,} \\ \lambda, \text{ if } n_1 = n_2 \text{ \& they are pre-terminal nodes,} \\ \lambda \prod_{j=1}^{nc(n_1)} [1 + C(ch(n_1,j), ch(n_2,j))], \text{ else.} \end{cases} \qquad (4)$$

Where, $nc(n)$ is the total number of children of node n and $ch(n, j)$ is the j-th child of node n in the tree. $n_1 = n_2$ denotes that the labels and production rules of node n_1 and n_2 are the same, and $n_1 \neq n_2$ denotes the opposite.

4. OUR APPROACH

There are two major modules through which the user query passes before the answer to the best matching question is sent to the user. Module 1 is responsible for finding the variants of the SMS terms in the domain and synonym dictionary and subsequently rank the questions based on the scoring function given in Section 3.1. Top scoring 100 questions are then passed to the Module 2 which re-ranks them based on their syntactic (Section 3.2) and semantic similarity. We give a stepwise description of the working of Module 1 in Section 4.1. Section 4.3 discusses how Syntactic Tree Matching (STM) and Wordnet based similarity measure (WN) can be used to improve the results. In the Intermediate Section 4.2, we explain the additional information exchange that take place between the two modules.

4.1 Variant Search & Bag of Words Retrieval

Variation matrix is generated based on the similarity measure given in Section 3.1. We redefined the Longest Common Subsequence Ratio (LCSRatio) of two strings as the ratio of the length of their LCS and the cube root of the average length of the two strings.

$$LCSRatio(t,s_i)=\frac{length(LCS(t,s_i))}{\sqrt[3]{avg_{l}ength(t,s_i)}} \qquad (5)$$

Cube root of the length is taken in order to give more weight-age to longer matching strings. In the original definition of LCSRatio by [8] length was used in the denominator as stop words were not considered in similarity measurement and hence all remaining were considered important irrespective of their lengths. However, we consider stop words also in the score measurement as they are important for syntactic tree similarity in the next module. Additionally, instead of using the length of dictionary term in calculating the ratio we used average of the lengths of the two strings to give importance to both the SMS token as well as the domain term to decide the weight of the similarity measure. This takes care of the fact that a longer term s_i in SMS implies the importance that the user has given to it while typing the query. Now, consider the query *"rmdy 4 wtry itchi eyes"*. A list of variants derived from the domain and Synonym dictionary is generated for each term in the query. The term *rmdy* occurs as remedy in the synonym dictionary which makes its corresponding term "cure" appear in the variation list of term "rmdy".

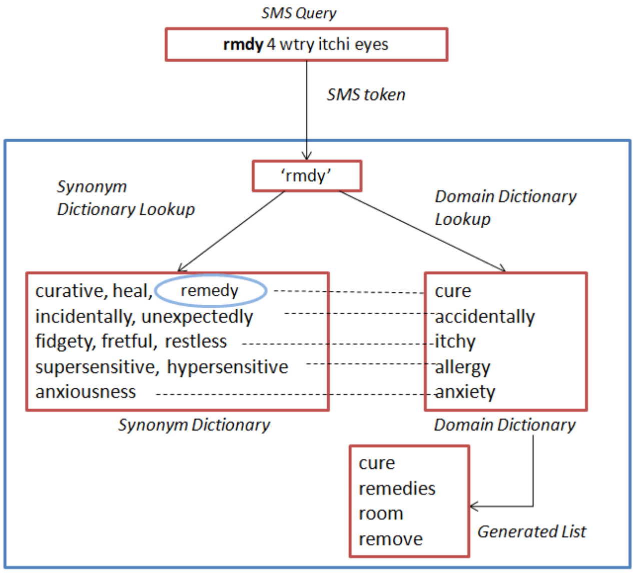

Figure 1: Synonym Dictionary Lookup for terms in the SMS Query "*rmdy 4 wtry itchi eyes*"(Based on Fig. 5 in [6])

In this way a variation matrix is generated with each column referring to the variants of a term in the SMS query (Figure 1).

Finally, the candidate set of questions from the corpus are obtained by using the scoring function given in Section 3.1.

4.2 Intermediate Step - Mapping the SMS query to the Questions in the corpus

In Module 2 (Section 4.3) the SMS query is syntactically compared with every question obtained in Module 1. Therefore the SMS query needs to be cleaned. We propose a solution in which a clean SMS query is generated for every question in R_BoW. Each term in the SMS is thus either replaced by its maximum scoring variant present in the question Q.

rmdy	4	wtry	itchi	eyes
cure	for	liquid	itchy	eyes
remedies	form	water	itch	eyelash
room	forgot	watering	it	even
remove	. . .	. . .	item	ever
. . .	. . .	. . .	. . .	. . .

Figure 2: Ranked List of Variations (Based on Fig.1 in [6])

Table 1: Example of Mapping noisy SMS query to FAQs

Candidate Questions	SMS^c
how to lose 20 pounds permanently in 4 months	how to lose w8 in few months
best way to lose 15-20 pounds in a month	Hw to lose w8 in few month
How many kgs a day to lose weight	How to lose weight in few mnths

The following formula describes the generation of clean SMS query. For every question Q in R_BoW SMS^c is defined as

$$SMS^c = s_1^c, s_2^c, s_3^c, ..., s_n^c \ where, \tag{6}$$

$$s_i^c = \begin{cases} t_m & \text{if } \alpha(t_m, s_i) = \max_{t_j \epsilon Q} \alpha(t_j, s_i) \ and \ \alpha(t_m, s_i) > 0 \\ s_i & \text{otherwise} \end{cases} \tag{7}$$

For example, SMS^c corresponding to different candidate questions for the SMS query "hw 2 luz w8 in few mnths" are given in Table 1.

In this way SMS^c is thus generated for all questions obtained from Module 1. In actual implementation this step can be performed along with question scoring step in Module 1. Now, SMS^c acts as the SMS query and is used in the next module.

4.3 Module 2 - Syntactic and Semantic Similarity (STM_WN)

Because of large number of lexically similar questions available in the corpus R_BoW gives high precision. However, finer analysis needs to be done in order to capture the syntactic and semantic similarity between the SMS query and the questions. By this, we conjecture that candidate questions obtained from BoW can be re-evaluated and the ones with more syntactic similarity with the SMS query can be moved up the ranking list. To achieve this, we employ a reformulation of the Tree Kernel method proposed by [11]. Node matching score can be formulated as the following recursive function:

$$M(r_1, r_2) = \begin{cases} 0, \text{ if } r_1 \neq r_2 \\ \\ Sem(r_1, r_2) * \delta_1 * \delta_2 * \\ \quad \lambda^{S_1+S_2} \mu^{D_1+D_2}, \text{ if } r_1 \text{ and } r_2 \text{ are terminals} \\ \\ \delta_{r_1}^{\eta} \delta_{r_2}^{\eta} \lambda^{2\eta} \mu^{\eta[2-(1+nc(r_1)(D_{r_1}+D_{r_2}))]} \times \\ \quad \prod_{j=1}^{nc(r_1)} M(ch(n_1, j), ch(n_2, j)), \text{ otherwise} \end{cases} \tag{8}$$

Where, S_i is the size of the sub-tree, λ is the size weighing factor, D_i is the depth of the sub-tree and μ is the depth weighing factor. λ and μ are two tuning parameters that denote the preference between size and depth. We set $\lambda = 0.1$ and $\mu = 0.9$ in order to give higher preference to bottom layer information. We redefined the node matching score for the case when the two nodes are terminal nodes as:

$$M(r_1, r_2) = idf * Sem(r_1, r_2) \delta_1 \delta_2 \lambda^{S_1+S_2} \mu^{D_1+D_2} \tag{9}$$

The Inverse Document Frequency was taken into account to prefer words that are highly discriminative i.e. words with a high idf score. Thus for example, matching of a term "Anaphylaxis" in the SMS is given higher weight than matching of the term "allergy", as in general a data set is expected to contain more questions containing the term "allergy" than the term "anaphylaxis". To measure the semantic similarity, $Sem(w_1, w_2)$ between two terminal words we used Leacock and Chodorow measure. Leacock's measure, uses the distance of the shortest path between two synsets to represent the semantic distance between two words. In order to scale the similarity metrics between 0 and 1, the following modified Leacock's version has been used:

$$Sem(r_1, r_2) = 1 - dist(w_1, w_2)/2D \tag{10}$$

Table 2: Search results for "home remedy for water itchy eyes"

Model	Questions(ordered by rank)
R_BoW	what is a home remedy for watery itchy eyes? home remedy for itchy, allergy eyes? any at home remedy for nasal drip and irritated sore/throat?
R_STM_WN	What is a home remedy for watery itchy eyes? home remedy for itchy, allergy eyes? **Best relief for itchy watery allergy eyes?**

Table 3: Search results for "does oats cause energy deprivation"

Model	Questions(ordered by rank)
R_BoW	Does sleep deprivation make your veins pop out more? How long does it take for scars from cutting yourself to go way? **Does oats cause energy loss?**
R_STM_WN	**Does oats cause energy loss?** Does sleep deprivation make your veins pop out more? Can allergies fade away like asthma sometimes does?

Table 2 and Table 3 show how this module brings potential similar questions in the Top 3 ranks and also improve the precision at top 1 respectively.

5. RETRIEVAL MODEL

Our retrieval model is inspired from the retrieval model proposed by [11]. We first index all the collected questions from Yahoo! Answers. By given a user query, an initial Noise Removal is carried out and Candidate questions are retrieved through what we call the BoW method. Top 100 of the initial retrieved results (R_BoW) are then selected and matched against the user query via the STM_WN module. A re-ranked matching result set (R_STM_WN) is thus obtained. Now, we are equipped with two types of relevance scores. Using only the R_STM_WN scores can be dangerous because low scoring questions in R_BoW have the tendency to move up the ranking list as the terms. Table 4 demonstrates this effect through an example.

Though the word "sunscreen" matches more with "sun-

Table 4: Relevance Ranking of questions for user query ”wt shud I do abt alrgy 2 sunscrn?” (a) Matching Questions in ranked order after passing thorugh Module 1. (b) Clean SMS query corresponding to each question in R_BoW (c) Reranked queries after analysis through Module 2

R_BoW(in ranked order)
What should I do about being allergic to sunscreen
What should I do about allergy to surgery scars

(a)

Candidate Questions	SMS^c
What should I do about being allergic to sunscreen	What should I do about allergic to sunscreen
What should I do about allergy to surgery scars	What should I do about allergy to surgery

(b)

R_STM_WN(in ranked order)
What should I do about allergy to surgery scars
What should I do about being allergic to sunscreen

(c)

scrn” the question containing ”surgery” moves up because of higher syntactic tree similarity. Therefore, our idea is to take into account both the relevance scores: one is the BoW score and other is the STM_WN score. We normalize the two scores before we employ a linear interpolation to combine them. The final ranking score used to obtain the final similar question searching result is as follows:

$$Score(Q) = \alpha NScore_{R\_BoW}(Q) + (1-\alpha) NScore_{R\_STM\_WN}(Q) \tag{11}$$

On Empirical testing we assigned value of 0.6 to α. $NScore_{R\_BoW}(Q)$ and $NScore_{R\_STM\_WN}(Q)$ are normalized scores of question Q in R_BoW and R_STM_WN resepectively. Figure 3 gives the overall architecture of the system.

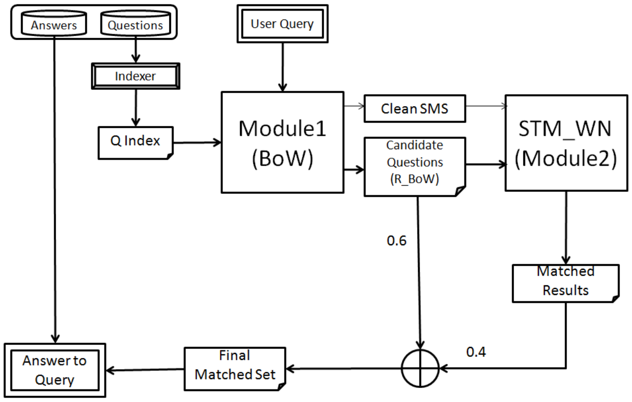

Figure 3: Overview of the Question Matching Retrieval System

6. EXPERIMENTAL SETUP

We validated the usability of our system by carrying out experiments on FAQ dataset collected from Yahoo! Answers. The FAQ Dataset consists of 7500 queries from three Yahoo! Answers categories namely Sports.Swimming, Sports.Tennis, Sports.Running. To measure the effectiveness of our system, we tested our system on the SMS query set collected from IBM, India Research Lab. The query set was obtained from the authors of [6] who generated it by asking human evaluators to choose questions randomly from the FAQ dataset. The evaluators typed the selected questions as SMS queries on a mobile keypad interface. In Figure 4, we show the comparison of performance of various systems.

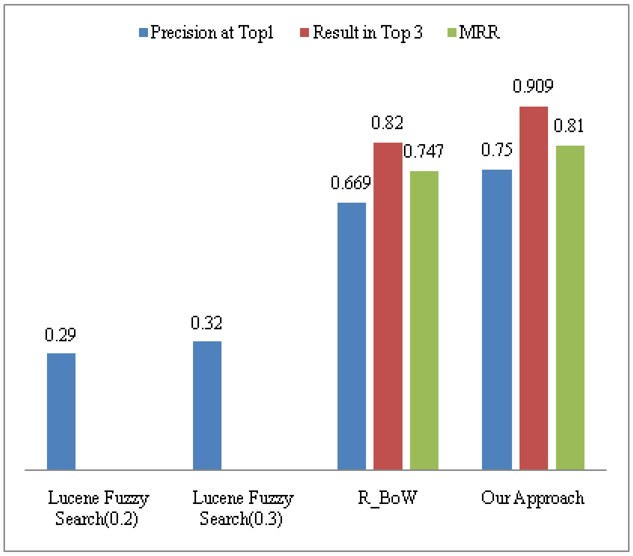

Figure 4: Performance measures over IBMs Sports SMS Query Set

Our system outperforms by ∼ 9% for Precision at Top 1 and by about 8% for the query in top 3 results. Accuracy of 90% for the result to occur in top 3 is a significant improvement in the performance and makes a SMS based FAQ retrieval system even more feasible for real word applications.

7. USER SATISFACTION AND RESPONSE TIME

For an Information Retrieval system to be complete, other two factors user satisfaction and response time also need to be considered. We briefly discuss each of these factors here. User Satisfaction depends on whether the answer/ information sent to him solves his information need or not. Often in CQA systems people provide reference to other online sites or information sources as answer to the query. These references though helpful to internet users may not be of any use to SMS users. Hence, answers to the queries need also to be analyzed for their relevance and different answers can be compared to find out the best amongst them. The Health category of Yahoo! Answers itself consist of 6M queries, therefore it behooves us to consider whether the amount of processing that is being done is feasible on such a large dataset. Shorter response time is essential so that the user can query the system multiple times to get more information by putting different/alternative combination of the query terms in real time. To test the response time we simulated 0.8M queries by selecting 10 successive words from a collection of biomedical articles. We consider

Table 5: Run-time for Module 1 for 0.8M question dataset

Processor	1 Intel Dual Core Processor	2 Intel Xeon Processors
RAM size	2 GB	16 GB
Run-time	30s	4s

this dataset as a representative data set for the Yahoo Answer FAQs. Only Module 1 is dependent on the dataset size as module 2 runs over fixed number of queries. We used the pruning Algorithm proposed in [6] to find best matching questions in Module 1. The Pruning Algorithm queries fewer terms and thus performs significantly better than the naïve algorithm and also gives near constant runtime performance for queries of different length. But with the increase in size of the dataset (or question corpus) the response increases approximately linearly. Table 5 gives the run-time for retrieving matching results from Module 1 on machines of different configuration.

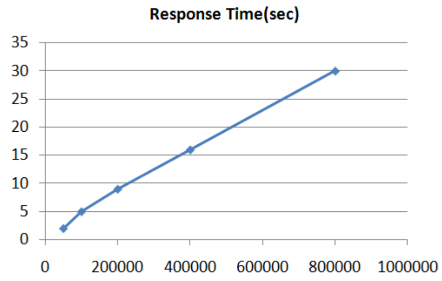

Figure 5: Response Time vs Size of the corpus(Number of Questions)

Given the response time of the order of several seconds, we believe that exploring distributed or other parallel computing options can allow the system to operate in real time.

8. CONCLUSION AND FUTURE WORK

Augmenting Noise Removal with IR techniques can improve the search efficiency. We showed how Syntactic Tree Similarity can be applied to noisy SMS queries. Our Experiments showed that there is an improvement of $\sim 9\%$ in precision at top 1. However, noise removal step is computationally expensive and behooves us to explore distributed options to put the system for real-world usage. Additionally, answer credibility for SMS users needs to be taken into account to further improve the user satisfaction in a real world setting. A host of future work ensues from this approach. Categorization information of CQA questions can be used for better precision. Other methods like vector space model and Okapi model can also be applied to retrieve matching questions for noisy queries. In SMS other kinds of noise are also present e.g. run offs ("how to" is written as "hw2"). We have not addressed this issue in this paper. By analysis of SMS text Levenshtein distance can be modified to give different weights to substitutions, additions, and deletions. Finally, answer analysis needs to be done for enhanced user satisfaction in a real world scenario.

9. ACKNOWLEDGEMENT

We would like to thank Govind Kothari, IBM India Research lab for his help in the development of the Noise handling part of the paper and providing us with the datasets to test our approach.

10. REFERENCES

[1] Apache lucene project. http://lucene.apache.org/.

[2] Extending google services in africa. http://googleblog.blogspot.com/2009/06/extending-google-services-in-africa.html.

[3] R. D. Burke, K. J. Hammond, V. A. Kulyukin, S. L. Lytinen, N. Tomuro, , and S. Schoenberg. Question answering from frequently asked question files: Experiences with the faq finder system. *AI Magazine*, 18(2):57–66, 1997.

[4] X. Cao, G. Cong, B. Cui, and C. Jensen. A generalized framework of exploring category information for question retrieval in community question answer archives. *The 19th International World Wide Web Conference (WWW)*, pages 201–210, 2010.

[5] S. K. Kopparapu, A. Srivastava, and A. Pande. In proceedings of the 4th international conference on mobile technology, applications, and systems. *In Proceedings of the 4th International conference on mobile technology, applications, and systems and the 1st International symposium on Computer human interaction in mobile technology*, 2007.

[6] G. Kothari, S. Negi, T. A. Faruquie, V. T. Chakaravarthy, and L. V. Subramaniam. Sms based interface for faq retrieval. *Proceedings: Joint Conference of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 852–860, Aug 2-7, 2009.

[7] K. Kukich. Technique for automatically correcting words in text. *ACM Comput. Surv.*, 24(4):201–210, 1992.

[8] I. D. Melamed. Bitext maps and alignment via pattern recognition. *Computational Linguistics*, 1999.

[9] E. Prochasson, C. Viard-Gaudin, and E. Morin. Language models for handwritten short message services. *In Proceedings of the 9th International Conference on Document Analysis and Recognition*, 2007.

[10] S. Riezler, A. Vasserman, I. Tsochantaridis, V. O. Mittal, and Y. Liu. Statistical machine translation for query expansion in answer retrieval. *In ACL*, pages 464–471, 2007.

[11] K. Wang, Z. Ming, and T. Chua. A syntactic tree matching approach to finding similar questions in community-based qa services. *In SIGIR*, pages 187–194, 2009.

[12] D. Zhang and W. S. Lee. Question classification using support vector machines. *In SIGIR 2003*, pages 26–32, 2003.