# Rule Based Approach for Word Normalization in Transliterated Search Queries

Varsha M. Pathak[*1], Manish R. Joshi[2]

*Research Coordinator, KCES's Institute of Management and Research, Jalgaon*

*Professor, School of Computer Sciences, Kavayatri Bahinabai Chaudhari North Maharashtra University, Jalgaon India*

***Abstract*** — SMS based Information Systems is the need of the age. Most of the present SMS based information systems send one way SMS based informative text messages generated from respective knowledge systems. By applying information retrieval methodology using models like Vector Space Mode, the systems can allow its users to send queries as per their requirement of information. This makes the system more fruitful from the user's point of view. This paper is about such initiatives for accessing relevant literature like poems, phrases, Rhymes, stories, abhang and much more. The mobile based quick library access system MQuickLib allows users to access such literature by formulating transliterated queries. The Vector Space Model is used to create the systems knowledge base by processing. The document terms and matched with the query terms by allowing variation in spelling due to transliteration style of the users. The matching score is assigned by devising a set of rules that identify the distance between two terms $d_k$ the term from document and $q_j$ the query term. The original Levenshtein's minimum edit distance algorithm is modified by applying this rule based approach. These rules are identified by collecting SMS queries from users for a given set of known queries in Marathi (Devnagari). Experiments were carried out for the collection of Marathi and Hindi literature that mainly include songs, gazals, powadas, bharud and other types. These documents are available in a standard transliteration form like ITRANS (an Indic Transliteration System). This paper elaborated a rule based approach and analyses the results to select appropriate rule based model that is further applied for the development of MQuickLib system.

***Keywords*** — *Information Retrieval; SMS Based Information System; Vector Space Model; Minimum Edit Distance; Noisy Query; Transliterated Search*

## 1. Introduction

Mobiles have activated and expedited the multi-modal communication technology. In modern age it has become an important gadget irrespective of the socio-economic boundaries, [10]. In mobile industry the Short Message Service (SMS) is the most reachable and affordable telecommunication service [2]. SMS is a cheaper and reliable way of text messaging tool which is universally accepted for sharing information, data and alerting instructions. The statistical data reports as available on [3], citizens of United States sent 69,000 texts every second in 2012" [CTIA 2012]. Similar scenario is observed in most of the developing and developed countries. According to a report of [Connect Mogul], only 43% of smart phone users make calls where as 70% users generally text. It gives convenience to communicate with people even if they are not available to answer immediately. Another interesting point is of response time. According to [CTIA 2010], on an average 90 minutes is the response time for an email, where as a text is answered by the recipient in only average 90 seconds.

Telecom Regulatory Authority of India (TRAI) has recognized Mobile based Value Added Services (MVAS). We referred the guidelines of TRAI on MVAS. TRAI recommends that SMS based Person to Application communication can turn into number of fruitful MVAS. This includes M-Governance, M-Education, M-Banking, M-health as major application areas. ASSOCHAM had forecasted that by the year 2015, in India alone the market of MVAS will reach up to Rs. 482 billion and large part of it can be accommodated by non-voice (text/images) messaging [1].

### *1.1 SMS based Information Retrieval*

Understanding the need of an SMS based information retrieval system a Mobile based Quick Reference Library System named as MQuickLib has been developed. The users can retrieve literature information like "Lata ne gayaa huaa shole film kaa gana", "Kavi ga di madgulkar yanni lihileli pawasvaril kavitaa". The conventional information retrieval methodology is applied with additional constraint to allow users to formulate their queries by using Short Message Service method. The idea is to allow people to send SMS based queries to retrieve required information.

The underline literature shows that SMS based Frequently Asked Question Answer and SMS based Natural Language Flexible Query processing are the important research extensions to the existing Information Retrieval. The wide coverage of Mobiles at grass roots of the society demands to explore Information Retrieval models with anew personalization dimension. Thus

MQuickLib is a prototype model that allows users to formulate queries in their natural language. To overcome the scripting problems of languages across the globe, transliterated text is allowed to the users. The users can spell the words of their language in Roman script. We have trained and tested the system on Marathi and Hindi languages. The related problems are investigated with a systematic development of an experimental model. The experiments conducted have revealed number of problems. The problems like dealing with transcription ambiguity and Named Entity Recognition (NER) ambiguities are the major issues those are investigated in the research undertaken by us. The experiments show the need of appropriate feedback mechanism for improving relevance of the answers produced by the system with the intention of users expressed in the respective queries. The query formulation style, transliteration style varies from users to users. The spelling could vary due to use of acronyms, abbreviations, synonyms and shortcuts used while creating the query terms. The style of creating query varies by applying Short Message Texting notations. The intentional word $d_k$ need to be guessed for a user query term $q_i$ need to be guessed by the system by applying suitable term matching mechanism. An hybrid approach has been developed on the basis of naïve Information Retrieval model Vector Space Model (VSM). For improving the results appropriate Relevance Feedback Mechanism (RFM) has been developed in support of the basic VSM model. The ambiguities due to transcription errors while spelling the terms by users could be resolved by applying a set of rules is the basic idea of this development. Following subsection focuses on the transcription ambiguity.

### *1.2 Transcription Ambiguity*

Transcription ambiguity occurring in user's query due to flexible styles of transliteration is the upcoming problem due to popularization of SMS in native languages. Many users use their own style for formulating the SMS query in their native language for information retrieval.

In our problem in order to use transliterated queries for Marathi Literature access on Mobiles we need to handle transcription ambiguity. The end users apply non-standard transliteration while compiling their information search queries in their own language. They use assorted language to formulate their queries.

This paper is the discussion about the work on SMS based Literature Information System as the extension to digitalization of present library facilities. This service needs to resolve the noisy terms to possible normal terms those occur in the systems vocabulary. We identify this as a word normalization problem. The systems vocabulary thus needs to be constituted of the normal terms occurring in the literature. In our problem we have constructed the vocabulary by processing the ITRANSed Literatureof Marathi language available on the internet sites like https://www.sanskritdocuments.org and http://www.giitaayan.com etc. This vocabulary of Marathi literature words extracted from these available sources is defined with a customized data structure that represents the Vector Space Model [11].

### *1.3 ITRANS Documents*

In MQuickLib system users are allowed to use their own transliteration style to spell the queries in their mother language (Marathi/Hindi language in this system) originally in its own script (Devanagari in this case) in Romanized form. Devnagari is the natural script of Marathi/Hindi languages, whereas people use English alphabets to spell the Marathi/Hindi words while interacting with others on computer network or on mobile network. Thus, we presume that users enter roman scripted strings to prepare Marathi / Hindi queries to access literature information. The transliterated terms written in the literature available at source are considered as the standard terms. Most of these documents apply ITRANS which is a standard transliteration style. These ITRANSed documents are processed to build the knowledge base using Vector Space Model. Respective algorithms are developed in Java using appropriate data structures.

To permit flexibility in Romanization of literature queries, the system has to improve its ability to recognize the query terms properly by matching them with most applicable standard vocabulary term which we term as "ITRANSed Normal terms". This problem is considered as the spelling correction problem. If a term t is spelled incorrectly it may not occur in system's vocabulary V, such term is defined as noisy term. We should map this noisy term to a normal term that occurs in the system's vocabulary.

## 2. Problem Modeling

This problem is compared with the SMS based Frequently Asked Question Answering system attained by many researchers. In respective published work on SMS based FAQ systems, the researchers apply new concepts like SMS query term normalization which is relevant to the language models. Significant research work is available related to this topic which was mainly assigned by Forum for Information Retrieval and Evaluation as one of the task [12].

The researchers [8] [9] have worked on this problem by applying Levenshtein's algorithm to compute distance between two strings in alphabetical manner. This Levenshtein's algorithm follows the dynamic programming method like Longest Common Subsequence.

DOI: 10.30726/ijlca/v7.i2.2020.72002

The important contribution of our model is that instead of normalizing the user query by using predefined set of Normal Queries in the system, our model normalizes individual terms by selecting a set of Normal Terms from the system's vocabulary. A minimum edit distance $t_i \simeq d_k$ is calculated by modifying the original Levenshtein's algorithm [8]. Here the terms t1, t2…ti are the terms in user query and d1, d2…..dk are the terms from document set D. The minimum edit distance is computed by applying this modified algorithm for each Noisy term $t \in Q$ and a Normal Term $d \in D$.

### *2.1 Minimum Edit Distance*

Edit distance between two strings is computed to measure the number of edit operations required to make the two strings exactly equal. The edit operations include insert, delete and substitute character in a string. A string x can be converted to another string y by applying different number of edit operations. For example RISK can be mistakenly written as MASK due to different editing errors. Similarly MASK can be corrected to intended word RISK by using different number of edit operations. These number of edit operations applied to correct the word w1 to w2 is known as w1-w2. The minimum number of edit operations that can correct the word w1 to w2 from all possible operations is known as minimum edit distance. Levenshteinv[9], has developed such algorithm by applying dynamic programming method of Longest Common Sequence finding problem. Some examples are shown in Figure 1 to explain the concept of different edit operations for conversion between different terms.

### *2.2 Query Term Normalization*

The basic Levenshtien's minimum edit distance function is modified by adding some rules to convert a corrupted term occurring in user query to its nearest standard tem occurring in system's vocabulary. The system's vocabulary is nothing but the inverse document term vector that we have generated from the Marathi and Hindi literature's document corpus. A set of rules are first identified on the basis or the query set collected from experiments carried with the help of graduate students. These students were given a set of queries in Devnagari which is the natural script of Marathi and Hindi. They were asked to rewrite these queries using their own transliteration style. A set of more than hundred and fifty such Marathi queries and forty Hindi song queries are used to train the system for identifying the term matching rules. A well defined set of these rules are then tested by the system to check the standard terms mapped for the user's noisy terms. In basic Vector Space Model, Term Frequency-Inverse Document Frequency (TFIDF) algorithm is used to assign matching weights to individual terms. The Query Term Normalization algorithm calculates a Proxy Weight scores for the query terms based on term normalization model. This mathematical model is expressed by expression Eq. 1 and its slight variation expressed by Eq.2.

For each pair of normal and noisy term, the basic formula is given in Eq. 1. As minimum edit distance between two terms measures how closer the two terms are, the ratio of the edit distance to the length of longest term quantifies the comparison of closeness of two or more terms with specified term. If a term $t$ is at edit distances $d_1$ and $d_2$ from two terms $x$ and $y$ with length $L_1$ and $L_2$ respectively then if $d_1/ L_1 < d_2/ L_2$ then $x$ is at closer distance from $t$ as compared to $y$ from $t$. With this mathematical rule the proxy weight of a term $v \in V$ quantifies its similarity with a user query term $t$ with Eq. 1. This formulation is modified to smooth the similarity measure in case if the distance is 0. This variation is given as in Eq. 2.

$$ProxWt_{t,v} = PrunWt - \frac{ed_{t,v}}{len} \qquad .... \quad Eq.1$$

$$ProxWt_{t,v} = PrunWt - \frac{ed_{t,v} - 1}{len} \qquad .... \quad Eq.2$$

Where,

- *PrunWt* is the term weight considered by applying a weight rules $R_w \subset R$
- *ed = minDist(t,v)* assumed by Levenshtien's minimum edit distance algorithm [9].
- *len* is calculated by term length rules $R_l \subset R$.

## 3. Rules and Models

Following preconditions are considered for the framework of the models varied by set of rules.

• The SMS query $Q$ is the sequence of $n$ number of terms $t_1$, $t_2$, .... $t_n$. It is clear that the query is noisy as one or more terms are distorted due to noise in transcription and SMS encoding mechanisms.

- $V$ is the vocabulary of the system that calculated by the Vector Space Mechanism on a set of literature documents. The terms $v_1$, $v_2$……$v_m$ are normal terms as they occurred in V.
- $ed = t_i \simeq v_j = minDist(t_i, v_j)$ is used to find minimum edit distance between the terms $t_i$ and $v_j$ .
- Let $W_1$ = 0:60, $W_2$ = 0:40, $W_3$ = 0:2, $W_4$ = 0:75 and $W_5$ = 0:25 are the weights used in the rules. These weights are computed for fine tuning theterm normalization. The weights are stabilized by adjusting them from 0 to1 through many experiments with the objective of correlated relevant terms with the noisy terms.

Following are the set of term length rules defined in the experiment. $L_1$ and $L_2$ are the lengths of the string $v_j$ and $t_i$ respectively.

Threshold condition is $ed < \alpha$ to calculate Proxy Weight for the term $v_j$, in association to query term $t_i$, the edit distance threshold value $ed(t_i, v_j)$ is equivalent to half of the length len, where it calculated by the rule set $Rl \subset R$. If edit distance is beyond the threshold value, the relevant vocabulary term $v_j$ is not in Candidate Normal Term set *NT*. The rule set for proxy weight estimation is on the basis of the length of tokens to match and some rules are explained for match making of tokens.

### *3.1 Comparison of Models*

We have identified various versions of the model by selecting diverse combinations of above rule set. The best performing four models on the basis of their Pruning Weight scores are selected for finalizing the matchmaking function of Proxy Similarity. We define them in the form of different models. We have selected four variations of these models to compare the performance based on precision and recall. The rules and their models are described below.

**Table 1: Rule set**

| **Length Related Rules** | |
|---|---|
| Rule 1 | ($l_1$>1 & $l_2$>1) shall be true for necessary precondition for proxy weight computation |
| Rule 2 | Calculate len = $l_1$>$l_2$ ? $l_1$ : $l_2$ |
| Rule 3 | For a vocabulary term $v_j \subset V$, if $l_1$> $l_2$ then len = $l_1$, and $l_1$> $l_2$ is important condition to qualify vocabulary term $v_j$ for Proxy Weight identification. |
| Rule 4 | Compute len = ($l_1$+ $l_2$)/2 |
| Rule 5 | Let length threshold $\alpha$ = *len/2* i.e. half thelen value the minimum edit distance then only consider the term for Proxy Weight calculation. |
| Character Matching Rules | |
| Rule 6a | if first characters of $t_i$and $v_j$match then flag$_1$ = 1, wf$_1$ = wt$_1$ |
| Rule 6b | if first characters of $t_i$and $v_j$do not match then wf$_1$=wt$_2$ |
| Rule 7a | if second characters of $t_i$and $v_j$match then wf$_2$ = wt$_2$ |
| Rule 7b | if second characters of $t_i$and $v_j$do not match then wf$_2$ = wt$_3$. |
| Rule 8a | if last consonant characters of the terms $t_i$and $v_j$match then wf$_3$ = wt$_4$. |
| Rule 8b | if last consonant characters of the terms $t_i$and $v_j$do not match then wf$_3$ = wt$_5$. |
| Pruning Weight = wf$_1$+wf$_2$+wf$_3$+wf$_4$+wf$_5$. | |

- Model 1 (M1): Basic Levenshtein's – This basic model assigns weights to the standard terms computed by applying Eq. 3. Here $ed_{t,v}$ is computed by applying Levenshtein's minimum edit distance between the noisy term *t* and standard term *v*. No other extra constraint is used in this model. Thus the pruning weight is set to 1. Similarly len is the length, computed asper the length specific Rule 2 and 5.

$$ProxWt_{t,v} = 1 - \frac{ed_{t,v}}{len} \quad .... \quad Eq.3$$

- Model 2 (M2): First Letter Match – In this model the terms are ranked on the weights by adding the constraints based on the weight rule set including rules 6a, 6b, 7a,7b which checks if first two letters match or not and assigns weights accordingly. Length factor (len) is calculated by applying rule 4 and 5.
- Model 3 (M3): Last Consonant Match- In this model in addition to first two characters the last consonant is matched. The pruning weights are calculated by applying rules 6a, 6b, 7a, 7b, 8a and 8b, Length specific rules (same as in Model 2).
- Model 4 (M4): Short Term Match - The Model-3 has proved better working on the sample data. It is further improved by restricting the length specific rules by applying rules Rule 3 and Rule 5.

## 4. Result Analysis

In this paper, the final model selection is made from evaluation of four versions in actual implementation of the system by applying standard measures. The experiments are performed on the word samples those show significant noise in the user queries. The Hindi dataset of 108 queries has generated 56 noisy words and in Marathi dataset, nearly 68 noisy words are produced from 152 queries. These noisy word samples are used in the evaluation of the above four normalization models. It shows a significant difference in mapping a noisy term to standard terms. The results are analyzed by applying standard measures as discussed below. Then, the best fit model is used to develop MQuickLib system.

### 4.1 Evaluation Measures

For comparison of these models four standard measures are used. The Mean Reciprocal Rank (MRR), precision for top 1 resultant term, top five resultant terms and top 10 resultant terms are calculated for all sample terms. An Average of each of these values computed on all sample terms shows the effectiveness of respective models. The Table-2 shows the results of the Average values of these standard measures on the sample data sets. The respective average values for the measures Avg_MRR_h, Avg_PH_1, Avg_PH_5 and Avg_PH_10 are for Hindi. Similarly Avg_MRR_m, Avg_PM_1, Avg_PM_5 and Avg_PM_10 are considered as the average values of the Marathi data. The graphical representation presented in Fig. 2 could be analyzed for the comparison of the performance of the models on the sample data sets. We can understand that the Models M3 and M4 are proved to be better performing than M1 and M2. For both these models MRR value is same i.e. 0.9166, as compared to 0.283 and 0.29 MRR values of M1 and M2 this score is comparatively high. The MRR value signifies the number of queries terms correctly normalized to top most term given by the system. M3 and M4 models have produced correct term at first rank in 91% cases where as M1 and M2 has produced in only 29% cases. Similar results we can see for other measures in Table-2. That means the models M3 and M4 could be selected for further development of MQuickLib.

## 5. Conclusion

The basic information retrieval models like Boolean model, vector space model, probabilistic models and combination of these models are applied in the problems of query based searching problems. The SMS based Information systems can be built by applying this theory which we identify as SMS based Information Retrieval. The Short Message Service is the cheaper and reliable feature of mobile systems. Many sectors like banking, healthcare, marketing and even governance are applying SMS as a tool of electronic messaging at cheaper cost due to wide coverage of Mobiles.

Considering the usefulness of this feature of telecommunication an initiatives are extended to use the SMS feature for more fruitful and interesting application area called as SMS based Information Retrieval. A Mobile Quick Reference Library MQQuickLib allows users to formulate their queries in their own language in Romanized (script of English) form. It also allows variation in transliteration and spelling variation. This is achieved by developing Query Normalization algorithm which applies a rule based approach to normalize a noisy term to its original word. The word intended by user is guessed by applying very effective minimum edit distance algorithm modified by applying some rules.

The term formulated by user by applying his/her own transliteration or spelling method is considered as a noisy term; whereas the term occurring in documents are termed as normal term. Mapping a noisy term to its nearest normal term is called as normalization. The rule based approach developed for query normalization has proved to be very effective in most of the cases.

The models M3 and M4 will be further experimented on wide range of queries to study their effectiveness and more effective model could be selected for the development of the system. Though this system is being developed for Marathi and Hindi, the methodology based on transliterated query search could be customized for other languages. For their respective transliteration style the rule set can be identified by applying the Query Normalization theory specified in this paper.

**Table 2: Evaluation of Models M1toM4**

| Measure | M1 | M2 | M3 | M4 |
|---|---|---|---|---|
| Avg_MRR_h | 0.28333 | 0.29 | 0.916667 | 0.916667 |
| Avg_PH_1 | 0 | 0.166667 | 0.833333 | 0.833333 |
| Avg_PH_5 | 0 | 0.458333 | 0.75 | 0.916667 |
| Avg_PH_10 | 0.091667 | 0.708333 | 1 | 1 |
| Avg_MRR_m | 0.8125 | 0.545 | 0.875 | 0.79125 |
| Avg_PM_1 | 0.75 | 0.375 | 0.875 | 0.75 |
| Avg_PM_5 | 0.79 | 0.77 | 0.79 | 0.76 |
| Avg_PM_10 | 0.80125 | 0.8275 | 0.84875 | 0.86 |

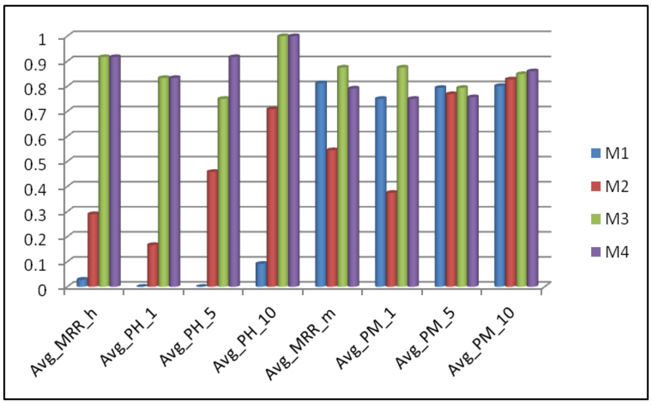

**Figure 2: Chart showing Average of Standard Measures on sample Data Sets**

## Reference

[1] Sanskrit documents collection : Home page, URL: http://www.sanskritdocuments.org,http://www.giitaayan.com/

[2] Rahis Shaikh Anwar Dilawar Shaikh, Rajiv Ratn Shah. SMS based FAQ retrieval for hindi, english and malayalam. 2013.

[3] CTIA annual wireless industry survey report. URL http://www.ctia.org/ industry-data/ctia-annual-wireless-industry-survey.

[4] Ian Ruthven and Mounia Lalmas. A survey on the use of relevance feedback for information access systems. The Knowledge Engineering Review, 18(02): 95–145, 2003.

[5] Pakray, Dr. Partha & Bhaskar, Pinaki. (2013). Transliterated Search System for Indian Languages.

[6] Sreangsu Acharyya, Sumit Negi, L Venkata Subramaniam, and Shourya Roy. Unsupervised learning of multilingual short message service (sms) dialect fromnoisy examples. In Proceedings of the second workshop on Analytics for noisy unstructured text data, pages 67–74. ACM, 2008.

[7] AiTi Aw, Min Zhang, Juan Xiao, and Jian Su. A phrase-based statistical model for sms text normalization. In Proceedings of the COLING/ACL on Main conference poster sessions, pages 33–40. Association for Computational Linguistics, 2006.

[8] Shahbaaz Mhaisale; Sangameshwar Patil; Kiran Mahamuni; Kiranjot Dhillon; Karan Parashar. Faq retrieval using noisy queries: English monolingual sub-task. In FIRE 2013 Shared Task on FAQ. DTU, Delhi, India, FIRE 2013, 2013.

[9] Govind Kothari, Sumit Negi, Tanveer A Faruquie, Venkatesan T Chakaravarthy, and L Venkata Subramaniam. SMS based interface for FAQ retrieval. In 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, pages 852–860. Association for Computational Linguistics, 2009.

[10] G Raghavendran. SMS based wireless home appliance control system. In Proceedingsof International Conference on Life Science and Technology (ICLST 2011), 2011.

[11] Pathak, V. M., Joshi, M. R. (2015). Natural Language Query Refinement Scheme for Indic Literature Information System on Mobiles. In Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India CSI Volume 2 (pp. 145-156). Springer, Cham.

[12] Majumder Prasenjit, Mitra Mandar, Agrawal Madhu, Mehta Parth (2015), Proceedings of the 7th Forum for Information Retrieval Evaluation. 10.1145/2838706.

DOI: **10.30726/ijlca/v7.i2.2020.72002**