International Journal of Computational Intelligence and Informatics, Vol. 4: No. 3, October - December 2014

Relevance Feedback Mechanism for SMS based Literature Retrieval in Indic Languages

Varsha M Pathak
Department of MCA
Institute of Mgt & Research North Maharashtra University, Jalgaon, M S India
pathak.vmpathak.varsha@gmail.com

Manish R Joshi
School of Computer science
North Maharashtra University, Jalgaon
M S India
joshmanish@gmail.com

***Abstract*- The concept and realization of 'Information Pulling' on handheld mobile devices facilitated an easy and effective information access. The researchers and developers are applying their efforts to concur mobile handsets into a timely business processing and information access terminals. Many of these applications use fixed format query answer method. This popularly used application type sends information pushing messages to mobile subscribers related to a specific domain. Agricultural messages, banking messages, health care messages, astronomy messages and advertising messages are examples of these types. This type of system is categorized as "Service Initiated Communication" system.**

Another category applies an information retrieval methodology for pulling information on mobile handset from the service server as per user's demand. This type of system is known as "User Initiated Communication" system. In this case the demand of information is either in fixed form or flexible form queries. Flexible form SMS query in natural language format for information access can be considered as recent research domain in this regard.

We have developed this second type of SMS based information system for Indic Language Literature. The system applies Vector Space Model for a suitable knowledge representation and an appropriate query-document similarity mapping scheme. In addition, we have focused on the development of a relevance feedback mechanism in order to improve the relevance of the responses of our system. Storage structure of VSM is tailored to represent the specified feature of Self Tagging Indic Literature Documents. The informative tags in documents, content, terms and the respective location of the terms are stored in the document's term vector. This paper elaborates this modified VSM structure and the relevance improvement mechanism in detail. The results are analyzed and discussed using Discounted Cumulative Gain.

Keywords-Information Retrieval, Relevance Feedback Mechanism, Probabilistic Model, Vector Space Model, Discounted Cumulative Gain

I. INTRODUCTION

Today's generation is using mobiles like a lifeline. Songs, Games, Pictures, Meaning of a word, Definition of a term, Recipe of a delicious dish, all this information is available on their smart phones. 3G, 4G technology has narrowed the differences in personal computers and mobile phones. The Operating environment of modern mobile systems is advancing with enhanced internal, external memory, high data rates, enhanced bandwidth and multimedia operating systems. The web based information systems are now accessible on these smart phones.

SMS can be considered as the basic feature of mobiles. The number of most fruitful applications, are under development. Our survey shows that this expansion is of two major types, GSM based automated remote controlling systems and SMS based information systems. SMS based information systems are again could be categorized as "Service Initiated Communication" and "User Initiated Communication". Many applications in agriculture, education, medicine, tourism, government, banking sectors, mostly are of the first type systems. In this type of service, the service sends a message that advertises the information available and prompts the user to access the information. These SMS in specified format carries the query and in turn user receives required information. This type of information systems generally forces users to use the service using "Pushing Information" strategy. On the other hand "Pulling Information" strategy applies "User Initiated Communication". In this case the user initializes the information accessible by sending SMS query either of "Fixed Format" or "Flexible Format". The system categorization and their functional taxonomy is discussed in detail in [15]. Following this survey we have developed our model for 'SMS based Literature Information System Indic Languages applying RFM' as shown in Fig. 1.

ISSN: 2349-6363

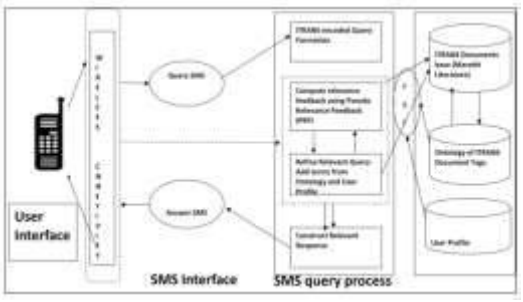


Figure 1. SMS based Literature Information System

This literature survey reveals that "SMS based information systems using natural language query" is a dynamic and challenging research extension to the existing Information Retrieval concept. Most of this development is related to Information Systems in English language. No significant work is reported in the field of "Literature Information Access in Indic languages on mobiles". Such information system can answer queries like "Who is author of certain poem?" or "What is the title of a song sung by certain singer of certain film?". Understanding this we have developed a SMS based Literature Information System for Indic languages.

For the implementation of an SMS based information system, we studied the underlined theory related to the conventional Information retrieval theory [2] [3] [4]. In our work we have augmented the Vector Space Model (VSM) with Implicit and Explicit Relevance Feedback Mechanism to improve user satisfaction regarding the search results. The system uses Client-Server paradigm where an Android Client communicates with Java Servlet. Literature documents encoded in ITRANS format are collected as the knowledge base at server side. ITRANS is one of the systematic transliteration method specially developed to interconnect Indic Languages [20].

This paper presents the VSM model specifically tailored to implement Self Documented Literature documents in Marathi and Hindi languages in transliterated form. The relevance feedback mechanism is designed by us to improve the result of this basic model. This enhanced model is elaborated in this paper. In the second section, we describe the methodology used for the development of our system. In third section two steps of relevance improvement applied in our project is explained. The system performance is measured and explained in fourth section. Analysis and results are presented in the conclusion section of this paper.

II. METHODOLOGY

Our system is SMS based Literature Information System (SMSbLIS). We are using literature of Marathi and Hindi languages as sample knowledge base of our system. We have chosen ITRANS transliteration method to represent this knowledgebase. This section describes the basic model, the related data structure used for implementation of SMSbLIS system. Result obtained and its analysis is presented in consecutive subsection.

A. Customised VSM Model

As explained by Salton [6] [7], we have build our basic model by applying Vector Space Model (VSM). Literature documents are preprocessed to extract set of terms that carry keywords germane to respective document as discussed by Van Rijsbergen [12]. A key term may relate to more than one document. This results in a key-term vector that germane to the whole document space. Similarly the query submitted by a user is processed to extract query term vector. The angular similarity between the Document Term Vector and Query Term Vector is computed by Cosine Correlation as expressed in "Eq. (1)".

$$\cos(doc_i, query_j) = \frac{\sum_{k=1}^{n}(term_{ik}, qterm_{jk})}{\sqrt{\sum_{k=1}^{n}(term_{ik})^2 \sum_{k=1}^{n}(qterm_{jk})^2}} \quad (1)$$

Where doc_i is the i^{th} document of the corpus , $query_j$ is the j^{th} query , $term_{ik}$ is the term weight of k^{th} term in i^{th} document, $qtermjk$ is term weight of k^{th} term of query j. The weights of terms are calculated using Term Frequency/ Inverse Document Frequency (TF-IDF) formulation

For a given query each document is assigned a cosine score. The VSM theory says that higher is the score higher is the relevance. Thus documents are arranged in decreasing order of their cosine score to obtain document ranking. We assume that the top ranked documents are precise to obtain required information. Table 1 depicts result of a query "s d burman ne sa.ngeet diya aur lataa ne gaayaa huaa gaanaa". The query is processed to remove stop words 'ne', 'aur', 'huaa' and query term vector is formed. The next subsection elaborates the data structure used for this model.

B. Storage Structure

To implement our model we have used JAVA platform. Let D= d1, d2, ...dm is the document space and T = t1, t2, ...tn is the key-term vector. A term ti may occur 0 to k times in any document di ϵ D. We used two HASHMAPs as data structure to represent this Document Term Vector Space (DTVS). The tagged lines of literature documents are processed to add both "Tag Term" and "Content Term" in DTVS. In first HASHMAP, each term ti is mapped to a unique code X. X is generated as next entry number in HASHMAP . Each term ti thus when occurs first time in document space D, is added in first hash map as < ti , S >. Where, S is a string. With this a new entry is also added in second HASHMAP as <ti, X>. Where X is the unique identifier of ti . The string S associated with term ti carries information of each occurrence of a term ti. Each occurrence of the term in any document is represented as a triplet <d,j,l>. Where this triplet points to the document d, tag term j with which the term ti is associated, and the respective location (line number) l. This triplet is appended in the string S at each occurrence of ti. This data structure thus produces inverse document index. We use this index to search terms occurring in respective documents, their attached tags and line number where term occurs.

Table-2 demonstrates the logical design of the data structure used to represent the document term vector. We have used two HASHMAPs as physical storage for this data. The table depicts an example. In this example five terms are included in term vector. These terms 'singer' , 'lataa' , 'starring' , 'kishora' and 'hema'. These terms are given identifier 1, 2, 3, 4 and 5 respectively in HASHMAP-1. HASHMAP-2 holds the occurrence information of each term. For term 'singer' the first triplet is '1,0,5'. The first value d=1 in this triplet indicates that the term occurs in document number 1. As it is a tag term itself its associated tag term number j=0. And the third value l=5 means that the term occurs on 5th line in that document. The figure shows that the term occurs in three more documents 2, 5, and 8 at 9, 7 and 7 lines in respective documents.

Let us look at other terms. The term 'lataa' occurs in documents 1, 5 and 8. It is associated with tag term 'singer', in all three occurrences. The term 'kishora' occurs in document number 2, location 8 as 'starring' (id=3) and in document 8, location 7 as 'singer' (id=1). Term Hema occurs as content term in document 5 as starring (tag no. 3) at location 6.

C. Document Tagging Feature

The result of VSM based system can be considered as fairly satisfactory. Relevant documents are top ranked for most of the queries. But as relevance between query and answer given by an Information Retrieval system is subjective, we found differences in judgments obtained from. In our example as above top ten documents have same similarity score (0.89). We received judgment of a expert. He has given positive remark to only five documents out of first ten documents. For example Doc#63 (Table-1) at 3rd position has received less relevance remark.

To solve this problem we studied the semi structure scheme of ITRANS documents. We found a set of tags are attached with the contents that appear in these literature documents. We shall standardize these tags to bring uniformity in identification of different type of information related to the literature. As literature category, language, title, writer, editor, date of compilation, date of creation need to be documented using unique tags. Different parts of the contents like 'Prakaran' (Chapter), 'Pada' (Stranza) are also could be tagged. We call these tagging words as "Tag Terms". Remaining tokens are called as "Content Term".

When we explored the results, we found that there is need to consider the semantic relation between the tag terms and content terms. Like <singer:lataa> is different than <singer:lataa, kishora>. Similarly <music: burman> is different than <singer:burman> or <lyrics:burman>. As expressed in "Eq. 1", our basic VSM model applies unigram term weighting method for cosine correlation calculation. Semantic relation between the words like 'gaayaa' (action,sing) and 'lataa' (person, singer) is not considered. Same applies to words sa.ngeet (music) and 'burman' (person, music director). Thus unigram tfidf weights count same similarity score for documents where the content terms 'burman' and 'lataa' occur.

In original queries, terms 'gaayikaa', 'sa.ngeet', 'geetkaara' are present. Whereas, in ITRANS documents, terms like 'singer', 'music', 'lyrics' are the synonyms (in English) of these words. These terms are the attributes on the respective contents. For example 'lataa' is name of a singer. The Hindi song documents has tagged line '\singer lataa'. Here 'singer' is tag and 'lataa' is its value. In user query, word 'gaayikaa' occurs which is Hindi synonym of 'singer'. We need to design a query refinement method to modify original query to add words like 'singer', 'music', 'lyrics' obtained from tags. Similarly remove words like 'gaayikaa', 'sa.ngeet', 'geetkaara' if they are unidentified. The words with lower tfidf weights are recognised as unidentified words. In our problem we considered the lower bound as zero in our example.

TABLE I. RANKED DOCUMENTS WITH COSINE SCORE

Query Keywords **1:S 2:D 3:Burman 4:sa.ngeet 5:diya 6:Lata 7:gaayaa 8:gaanaa**		
Rank#	***Document#***	***Cosine Score***
1	19	0.8944271909999159
2	23	0.8944271909999159
3	63	0.8944271909999159
4	133	0.8944271909999159
5	207	0.8944271909999159
6	326	0.8944271909999159
7	341	0.8944271909999159
8	418	0.8944271909999159
9	422	0.8944271909999159

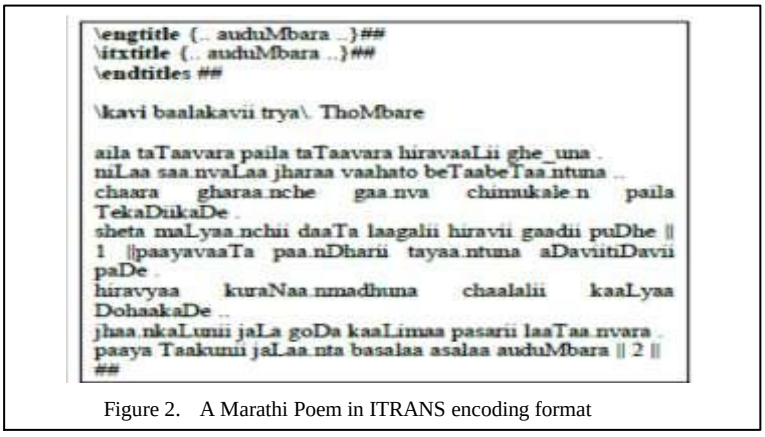
To explain the concepts of tag term and content term we present a well known Marathi Poem "audumbar" as ITRANS formatted document in Fig. 2. In this document we see tag terms and content terms as given in following table. The tag terms are single word attributes where as associated content terms are multi-word values. For example 'kavi' (writer of poem) is a tag term which is associated with content terms 'baalakavi' , 'trya\.' , 'ThoMbare'. We considered these content terms as separate strings and hence separate keywords in term vector. In addition we see '||1||', '||2||' are the end of first stanza, second stanza respectively.

TABLE II. LOGICAL DESIGN OF DOCUMENT TERM VECTOR

Term id	**Term Vector Details**				
	Term	***Term Type***	***Document Vector in string of triplet format***		
1	singer	Tag	1,0,6	2,0,9	5,0,7…… 8,0,7 …….
2	lataa	Content	1,1,6	5,1,7	8,1,7……..
3	starring	Tag	1,0,5	2,0,8	…… 5,0,6
4	kishora	Content	2,3,8	8,1,7	…….
5	hema	Content	5,3,6	…..	…….

III. RELEVANCE IMPROVISION

There are number of relevance improvement techniques. User query is improved by query refinement using techniques like ontology. In our system we have designed 'literature information ontology'. Another way is using user's explicit or implicit relevance feedbacks. The relevance feedbacks are also used to clear query vagueness and make it more precise. This is definitely going to improve system's responses. Most of these techniques are based on Rochchio's relevance feedback model[17] or Bayes probabilistic model [12]. We focus to blend Information Retrieval with Relevance Feedback Mechanism (RFM) suitably for mobile users.


\engtitle {.. auduMbara ..}##
\itxtitle {.. auduMbara ..}##
\endtitles ##

\kavi baalakavii trya\. ThoMbare

aila taTaavara paila taTaavara hiravaaLii ghe_una .
niLaa saa.nvaLaa jharaa vaahato beTaabeTaa.ntuna ..
chaara gharaa.nche gaa.nva chimukale.n paila TekaDiikaDe .
sheta maLyaa.nchii daaTa laagalii hiravii gaadii puDhe ||
1 ||paayavaaTa paa.nDharii tayaa.ntuna aDaviitiDavii paDe .
hiravyaa kuraNaa.nmadhuna chaalalii kaaLyaa DohaakaDe ..
jhaa.nkaLunii jaLa goDa kaaLimaa pasarii laaTaa.nvara .
paayaa Taakunii jaLaa.nta basalaa asalaa auduMbara || 2 ||
##

Figure 2. A Marathi Poem in ITRANS encoding format

We design our SMSbLIS model taking advantage of mobile technology for explicit and implicit RFM. While mobile interface is under development to obtain explicit and implicit feedbacks from users, our experiments involve few experts to obtain impartial judgments. We use these judgments as explicit feedback. Using this feedback probability relevance model is applied in our system.

A. XML Tagging

Numbers of projects are undertaken by developers and researchers using XML tagging for indexing purpose. As said by Erdmann, Studer [21], Extensible Markup Language (XML) is used concur the information of real word with semantic realization. XML can be used for knowledge building and its dissemination. We explore this strength of XML tagging for connecting semantically similar words in natural language text of Hindi and Marathi literature documents. We have developed XML tagging for our literature tags. These tags are in English language and are mapped to the respective synonyms occurring in the query of Indic languages. We have build this XML tagging using sample queries that we have collected from our experiments. The semantic disambiguation is resolved by adding relative verbs occurred in these sample queries. Fig. 3 is the sample of this XML tagging that we propose to extend further into a systematic Ontology. For example a words like 'gAyale' (sung, verb) and 'gAyikA' (singer, noun) relates to tag 'singer'. Here the tag 'singer' occurs in the literature document and the verb 'gAyale' , noun 'gAyikA' occurs in the query.

Applying this now a query of our example is modified into "s d burman lata music singer title". The scheme removes terms 'sa.ngeet', 'gaayaa' , 'gaanaa' and adds tag terms 'music', 'singer', 'title'. These terms are obtained from 'literature information ontology'. As these tag terms occur in all documents of same literature type their inverse document frequency is computed as zero. Thus addition of these terms has no effect on Cosine score of respective documents. We need a better scheme to assign relevance with respect to the <tag term: content term> pair values. The method need to have different relevance score to documents with <singer:lataa> than documents with <singer: lataa, kishora>. The documents with <music:burman> should have different relevance score than documents with <lyrics: burman> or <starring: burman>. With this consideration we build the probability relevance model as described in next subsection.

B. Probability Relevance Model

If we have some document D and query Q, we have two events:

1. L, is the event that D is liked or relevant for a given query q.
2. L', is the event that D is not liked or not relevant for a given query q.

Here we consider that each document is described by a set of attributes, value pairs. The tag terms are the attributes and the content terms are the values of the attributes. In implementation of probabilistic relevance model of our system, we are interested in these <Tag, Content> pairs. Instead of unigram terms we now consider bigrams in the form <ti,tj>, where ti , tj are two terms occur in same document and are semantically dependent.

The semantic relationship between 'tag term' and 'content term' is conveyed by 'XML tagging' of Literature Information System'. Occurrences of the bigrams in the document set are computed. To gain relevance feedback, expert users are asked to judge document d Є D to be relevant or not relevant. As the response has only two values it is known as Binary Relevance Feedback (BRF). This relevance feedback of top ranked documents is used to estimate probability of relevance of next documents for respective query. The probability relevance mechanism is based on following.

Let P(L|D) , is the probability that a document with description D is relevant. The description D indicates presence and absence of the <ti , tj > pairs in the respective document using Vector Space Model of our system.

Given the estimated probability P(D|L) that is the probability that a document is relevant for the given query Q. If prior probability of any document being relevant is P(L) and the probability that a document D is observed independent of its relevance is P(D) then,

The Probability Model follows Naive Bayes' theory [13]. We calculate P(L|D) by applying Bayesian inversion formulation as shown in "Ex. (2)" as discussed by Ruthven [13] .

Let P_{ik} is $P_q(x_i = 1| L)$ that is the probability of relevance of the document d_k having term xi for given query qm . Q_{ik} is $P_q(x_i = 1| L)$ probability of non relevance of the document d_k having term x_i for the query q_m . We derive equation 3 using independence assumption 2 and ordering principle 2 as discussed by Ruthven in [13].

Where C_{ik} is the weight of individual term calculated applying equation 3. Here T is the term vector. Thus according to equation 4 RSV is calculated as the sum of cost of terms t_i occurring in a document d_k relevant to the query q_m. P_{ik} and Q_{ik} are estimated by applying relevance feedback as described below. One can understand that the probability weight C_{ik} for all term pairs $<t_i\ t_j>$ and documents d_k not only depends on presence of the term but also on absence of the term.

```
<Literature>
 <LitSection value="10">
   <SecName>Poetry</SecName>
   <Language>Hindi</Language>
      .
      .
      <Tag>
            <Tname>singer</Tname>
            <Type>person</Type>
          <Synonym>gaayak|gaayeeka|gAyaka|gAyeekA</Synonym>
              <Verb>gaayaa|gAyA</Verb>
              <PreSymbol>slash</PreSymbol>
            <Endswith>hash|percent|dollar</Endswith>
          </Tag>
      .
      .
      .
 </LitSection>
</Literature>
```

Figure 3. XML Tagging Sample for ITRANS Documents

As said above we asked experts of domain knowledge to give binary (yes for relevant, no for not relevant) judgments for top ten ranked documents of the VSM module. From this judgment remaining documents are judged applying the probability weights of individual <Tag, Value> pair. Documents are ranked based on Retrieval Status Value (RSV) as in equation 4 applying "Probability Ranking Principle" stated in [11].

$$P(L|D) = \frac{P(D|L)P(L)}{P(D)} \tag{2}$$

$$C_{ik} = \log \frac{P_{ik}(1-Q_{ik})}{Q_{ik}(1-P_{ik})} \tag{3}$$

IV. Result Analysis

As discussed in this paper we have developed a method that not only refines given query as per the relevant tags using XML, but also improves systems ranked list. More relevant documents are promoted to higher position applying our model. We used standard measure, Discounted Cumulative Gain (DCG) to analyze system performance. These measures are useful to understand graded relevance for each rank (r). DCG cumulates the relevance gain of previous r-1 ranks to compute relevance gain of rank r. If a retrieval system has placed a document at position p then DCGp value indicates as per user's judgments whether it is suitable for that position with specified grade on not. We applied our relevance improvement model using number of queries of Marathi and Hindi Poems and lyrics of Marathi, Hindi songs. In this paper we present performance of the improved system by comparing DCG over five queries.

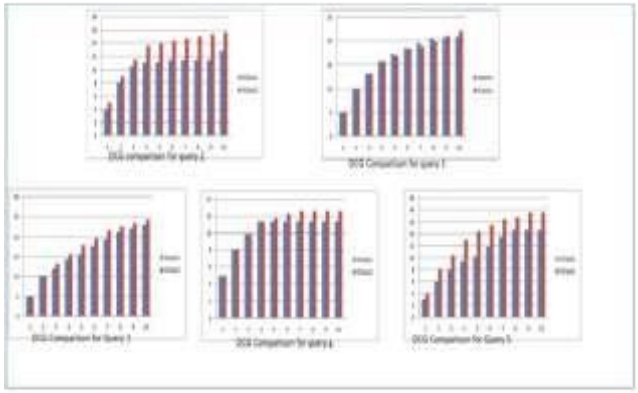


Figure 4. Discounted Cumulative Gain Comparison of top 10 ranks for 5

For each specified query q, we obtained DCGp for each position (rank) p applying five points grading of the top ten documents given by system. The point 5 denotes highest relevance. That means exact answerable document is observed. Point 0 means no relevance with the query at all. And the intermediate points 4, 3, 2, 1 respectively denote partial relevance with high, moderate, low and very low levels. The result of Discounted Cumulative Gain over top ten positions (ranks) for each of five sampled queries is depicted in Figure 4.

From above graphical representation we can observe a definite improvement in the performance of the system. In Figure 4 we present analysis of the performance of the system for each query individually. In Each chart denotes comparison of the DCG for each ranked position from rank 1 to rank 10. We could observe that

there is no noticeable difference in first 4 to 5 ranks. But next ranks are affected by getting more relevant document at that position. That means for all the sampled queries the relevant documents from top 25 documents have been rearranged by the system at appropriate places as compared to the basic model.

V. CONCLUSION

This paper is about development of our SMS based Literature Information System (SMSbLIS). The system expects an SMS query from user asking for literature information. The system has to respond with list of relevant document. The system is based on Vector Space Model (VSM) and has support of a suitable knowledgebase built on 'ITRANS formatted Indic Language Literature text documents'. The documents are self documented. We used this feature to refine user query to add more relevant words. We recovered these relevant words by mapping literature attributing words occurring in user query with the synonymies words occurring as self documenting tags in literature documents. This mapping is maintained by the system in the form of 'Literature Information Ontology'. Query refinement is the first step to enhance relevance of the search result. In second step our VSM based SMSbLIS model progresses to enhance user satisfaction level. For this the VSM based system is blended with Relevance Feedback Mechanism. We aim to use RFM by indulging benefits of mobile technology. Our efforts are initialized by implementing binary relevance feedback mechanism using Probability Model.

We used DCG as the standard measure to test improvement in the system performance. We found there is definite improvement in the user satisfaction. This result is graphically presented in this paper in Fig. 4. We are designing our next experiments to test system performance for more rigorous queries. We also plan to expand the document corpus for this experiment.

REFERENCES

[1] Sparck Jones K. : Automatic Keyword Classification for Information Retrieval. Butterworth's, London (1971).

[2] Luhn H. P. : The automatic creation of literature Abstracts. IBM Journal of Research and Development, vol. 2, pp. 159-165 (1958).

[3] Schultz C. K., H. P. Luhn: Pioneer of Information Science - Selected Works, Macmillan, London (1968).

[4] Porter M.F. :An algorithm for suffix stripping, Program. vol 14(3), pp. 130-137, (1980).

[5] Salton G. : Automatic Information organization and Retrieval. McGraw-Hill, New York (1968).

[6] Salton G. : Automatic text analysis. Science, vol. 168, pp. 335-343 (1970).

[7] G Salton (ed) :The SMART retrieval system – experiments in automatic document processing. (1971).

[8] C. Buckley, G. Salton, J. Allan and A. Singhal. : Automatic query expansion using SMART TREC-3. the Third Text Retrieval Conference (TREC-3). D. K. Harman (ed). NIST publication 500-225. pp 69-80. (1995).

[9] Sparck Jones K. : A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation, vol. 28, pp. 111-21 (1972).

[10] Yu C. T. and Salton G. : Effective information retrieval using term accuracy. Communications of the ACM, vol. 20, pp. 135-142 (1977).

[11] S. E. Robertson : The probability ranking principle in IR. Journal of Documentation,vol. 33. 4. pp. 294-304. (1977).

[12] C. J. Van Rijsbergen: Information retrieval. Butterworth's. 2nd edition, (1979).

[13] Ian Ruthven, Mounia Lalmas, 'A survey on the use of relevance feedback for information access systems', The Knowledge Engineering Review vol 18 Issue 2, June 2003 pp. 95-145, Cambridge University Press New York , USA.

[14] E. M. Voorhees and D. Harman : Overview of the fifth Text REtrieval Conference (TREC- 5). Proceedings of the 5th Text Retrieval Conference. pp. 1-29. NIST Special Publication 500-238. Gaitherburg.1996.

[15] Manish Joshi, Varsha Pathak . : A Functional Taxonomy of SMSbIR Systems. 3rd international conference on Electronics Computer Technology, Kanyakumari, 8-10 April, 2011, pp. 166-170 (2011).

[16] Ran A. and Lencevicius R.:Natural Language Query System for RDF Repositories. To appear in Proceedings of the Seventh International Symposium on Natural Language Processing, SNLP (2007).

[17] J. Rocchio. :Relevance Feedback in Information Retrieval , in Salton: The SMART Retrieval System: Experiments in Automatic Document Processing, Chapter14, pp. 313- 323, Prentice-Hall, (1971).

[18] Hassell J. Aleman-Meza. B. Arpinar. I.B.: Ontology-driven automatic entity disambiguation in unstructured text. International Semantic Web Conference, Springer pp. 44–57 (2006).

[19] Karanastasi, A., Christodoulakis, S.: Ontology driven semantic ranking for natural language disambiguation in the ontonl framework. In Franconi, E., Kifer, M., May, W., eds.: ESWC. vol. 519 of Lecture Notes in Computer Science. Springer pp. 443–457 (2007).

[20] Varsha Pathak, Manish Joshi . ITRANSed Marathi Literature Retrieval Using SMS based Natural Language Query. Advances in Computational Research, vol. 4 (1), pp. 125-129 (2012).

[21] Erdmann, Michael, and Rudi Studer. Ontologies as conceptual models for XML documents. Proceedings of the 12th International Workshop on Knowledge Acquisition, Modelling and Mangement (KAW'99), Banff, Canada, October, (1999).