IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.7, Special Issue.1, Jan 2019 E-ISSN: 2347-2693

A Comparative Study of Three IR models for Bengali Document Retrieval

Soma Chatterjee[1*], Kamal Sarkar[2]

[1] Department of Computer Science & Engineering, Jadavpur University, Kolkata, India
[2] Department of Computer Science & Engineering, Jadavpur University, Kolkata, India

[*]*Corresponding Author: somadey05@ mail.com, Tel.: 9903114503*

Available online at: www.ijcseonline.org

***Abstract*—**In this paper, we studied and examined some selected information retrieval approaches for Bengali information retrieval. These approaches used keyword to describe the content of each document. We choose three models to understand their working mechanisms and shortcomings. These models are TFIDF Vector Space model, Latent Semantic Indexing (LSI) model, and BM25 model. This understanding is important to overcome these shortcomings. These models are examined on our created Bengali dataset and Bengali queries and the results are stated in the result section in this paper. Our study reveals that Okapi BM25 model performs best among the three IR models studied for Bengali document retrieval.

***Keywords*—** Information Retrieval, Bengali language, LSI, BM25, probabilistic, Query.

I. INTRODUCTION

Information retrieval (IR) system is a task of retrieved relevant information from large information collection. But find the relevant information from the tremendous large amount of collection; the challenge is how quickly find the relevant information in response to a query. For a specific user query, the task of IR model is to calculate the score of a document which specify how a document is relevant to user query and returns set of relevant document based on score.
So, many different IR models have been developed in the earlier years but they are focused on English language. But we are trying to investigate, how different IR models works for Bengali document and Bengali queries. For our experiment we choose three models, these are TFIDF based vector space model (VSM), LSI model and BM25 model. Mean Average Precision (MAP) is used for evaluation of these model. These entire models used the same frame to represent query and document. They represent each document and query is a set of words.

But one of the fundamental problems in information retrieval is the word mismatch problem which arises from the fact that same question may be asked in different ways using different sets of words. It is also the fact that similar concepts may be presented in different ways in the different documents.

Section I contains the introduction of IR model, Section II contain the related work of IR model, Section III contain the explain the Methodology with, Section VI describes evaluation and results, Section V concludes research work with future directions).

II. RELATED WORK

The most common information retrieval models that are in use are Boolean retrieval model, Probabilistic model, Vector Space Model, Language model and LSI model.

The earliest works on Information retrieval model was devoted to Boolean Retrieval Model which is the simplest and most widely used model. The model relies simply on Boolean operators like AND, OR and NOT. The terms are linked together with these operators. However it searches the exact words thus a user has to have some idea about the query he is using, for example misspelling will not give an intended result. Disadvantage of these models is, it only concentrates only syntactic matching of word. To deal with this, the common strategy is to use Stemming which reduces a term to its morphological stem and using it as a prefix, users can retrieve many terms that are conceptually related to the original term [11].

There have been many attempts to help users overcome some of the disadvantages of the traditional Boolean discussed above. The Smart Boolean, was developed by Marcus (1991; 1994). It tries to help users construct and modify a Boolean query as well as make better choices along the several dimensions that characterize a Boolean query.

Many users, particularly professionals, prefer Boolean query models. Boolean queries are precise: a document either

matches the query or it does not. This offers the user greater control and transparency over what is retrieved. And some domains, such as legal materials, allow an effective means of document ranking within a Boolean model. However, this does not mean that Boolean queries are more effective for professional searchers. A general problem with Boolean search is that using AND operators tends to produce high precision but low recall searches, while using OR operators gives low precision but high recall searches, and it is difficult or impossible to find a satisfactory middle ground. Boolean queries just retrieve a set of matching documents, but commonly we wish to have an effective method to order (or "rank") the returned results.

Unlike Boolean model, Vector space representation is a different representation of documents which is used to classify documents and the representation is used in information retrieval systems[15][16][24]. Here each document is considered to be a vector (one component of vector for each term). Since terms are axes of high dimensionality, they are normalized to convert them to vectors of unit length. The standard TFIDF based vector space model (VSM) suffers from the word mismatch problem when the query words and document words do not exactly match [15][16]. Due to this problem, the traditional TFIDF based vector space model (VSM) gives poor recall. Stemming is usually used to improve the recall of the IR systems. Moreover, though stemming can alleviate from the problem posed by the inflectional word forms, it cannot handle the semantic level word match [7] [8][10].

Latent Semantic Indexing [4] is also a global technique that maps the high dimensional vectors corresponding to documents into a low dimensional space where the related terms which are orthogonal in the high-dimensional space will have similar representations in the low dimensional space, and as a result, retrieval based on the reduced representations becomes more effective. The similarities among the documents can then be estimated in the reduced space. This approach was shown to be very promising, especially at higher levels of recall [21].

Probabilistic model gives a relative ranking of documents based on "probability ranking principle"[14]. It is basically a statistical model of information retrieval. It utilizes frequency measures to determine relevance of documents with respect to queries. Given a query, the probabilistic model computes probability that each document is relevant to the query or not. Each document is described by the presence/absence of index terms in the document as binary vector. The traditional Bayes' Theorem is used to calculate the probabilities indicating degree of relevance of the documents with query. The query scores can be calculated by the different probability measures such as Tree-structured dependencies between terms [20], Okapi BM25 [17], Bayesian network approaches to IR [18][19] and many others.

The language modeling approach was first introduced by Ponte and Croft in (1998). A new way to score a document was done. It is known as the query likelihood scoring. It was proposed to consider a document to be a bag of words and whether a document can generate a query. If a document can generate a query then it can be said that the document is relevant to the query. Formally, the general idea of the query likelihood retrieval function can be described as follows. Let Q be a query and D a document. Let θ_D be a language model estimated based on document D. The score of document D with respect to query Q is defined as the conditional probability $p(Q|\theta_D)$. That is, $Score(Q, D)= p(Q|\theta_D)$.

Thus defining θ_D and estimating it with respect to the documents is the main challenge. The model θ_D is a multiple Bernoulli model. V is the vocabulary set of the language of the document set. A binary random variable Xi is defined for each word $wi \in V$ to indicate whether the word wi is present or absent in the query. Thus model θ_D would have precisely |V| parameters which can model presence and absence of all the words in the query.

According to this model, the query likelihood can be computed based on two types of probabilities-(1) the probability that a query word present in the document is generated by the document and (2) the probability that a query word absent in the document is generated by the document.

One problem with this maximum likelihood (ML) estimator is that an unseen word in document D would get a zero probability, making all queries containing an unseen word has zero probability, which is clearly undesirable. More importantly, when a document is a very small, the ML estimate is generally not accurate. So an important problem that is to be solved is to smooth the ML estimator so that we do not assign zero probability to unseen words and can improve the accuracy of the estimated language model in general[25].

The early IR research focuses mainly on development of IR techniques for English. Recently the interest in the development and automatic evaluation of information retrieval system for Indian languages is also growing. Sarkar and Gupta (2016) present a comparative study on the performances of various IR models for Bengali information retrieval.

Dolamic and Savoy [5][12] [13] evaluated the performance of various IR models for Bengali, Hindi and Marathi languages. Some approaches to Bengali monolingual retrieval have been presented in [1] [3] [6][9]. Though a number attempts has been made by the researchers to

develop IR system for Indian languages, they have mostly focused on enhancement of the traditional vector space model for IR system by improving the stemming process. Ganguly et al. (2013) deviates from this tradition to some extent by investigating the effect of decompounding for Bengali IR and Barman et al. (2013) performed Query expansion using Wikipedia and performed Entropy-based ranking.

In this article, we study on how the different IR models works and evaluate some selected IR model for Bengali document and query.

III. METHODOLOGY

A. TFIDF based Vector Space Model for IR

1) Vector Space Model (VSM)

The Bag-of-Words model, views a document/a query as a collection of words. Given a collection of documents C, containing words from a vocabulary V, the following information can be extracted from each document [15].

Term Frequency (TF): For a word, the Term Frequency measures the frequency of a word in document. We have used a modified form of TF as:

$$Modified\ TF \quad = \quad \log(0.5 + TF) \qquad (1)$$

Document Frequency (DF): For a word, the DF measures the number of documents in the collection C, the word is present in. DF is used to calculate the Inverse Document Frequency IDF, which is an important measure in IR.

Inverse Document Frequency (IDF): it is the inverse of the DF. So if a word is rare, it has a low DF, and its IDF is high, and if is present in a large number of documents in the collection, it has a high DF, and its IDF is low. IDF is calculated using the formula:

$$IDF(w_i) \quad = \quad \log\left[0.5 + \frac{N}{DF(w_i)}\right] \qquad (2)$$

Where: N is the number of documents in a collection.

Vector representation: As Bag-of-Words model represents each document and query as collection of words, each document and query is represented as vector of length v is the vocabulary size. Each component of the document vector or query vector corresponds to a word in v. a document

vector for d, $\vec{d}$ = (w_1, w_2, ….w_v) where w_i is TF*IDF weight of word x with the i^{th} index in the vocabulary and x is present in d. If the x is not present in d w_i is set to 0. TF and IDF are calculated using equation (1) and equation (2). Similarity, for agiven query q, we compute query vector $\vec{q}$ = (q_1, q_2,….q_v) where q_i is TF*IDF weight of the i^{th} vocabulary word present in query q. Here TF indicates the frequency of query word in the query q.

For the sake of computational efficiency, we use dot product of document vector and query vector as the relevance score instead of computing cosine similarity[15]. Relevance score for a document *d* is:

$$\text{TF} - \text{IDF Score(d)} = \sum_{w \in Q \wedge U \in d} TF - IDF(w) * TF - IDF(u) \qquad (3)$$

Since dot product becomes too large, we apply log normalization.

$$Modified\ TF - IDF\ Score(d) = \log\big(TF - IDF\ Score(d)\big) \qquad (4)$$

For the sake of computational efficiency, for given query, we consult inverted index to retrieve documents relevant with the query words at a time. While information of the relevant documents is extracted from the inverted index, TF-IDF information is also extracted. Since the outputs of the models are finally combined, their outputs should be properly normalized. So, we apply soft max function on the logarithm of dot product to normalize again. We have used the following Soft max function for this purpose:

$$Softmax\ normalize\ value(d) = \frac{e^{Modified\ TF-IDF\ Score(d)}}{\sum_{d \in D} e^{Modified\ TF-IDF\ Score(d)}} \qquad (5)$$

The result of equation (5) is also very small. So, we normalize again this value by using traditional min-max procedure.

B. LSI based IR Model

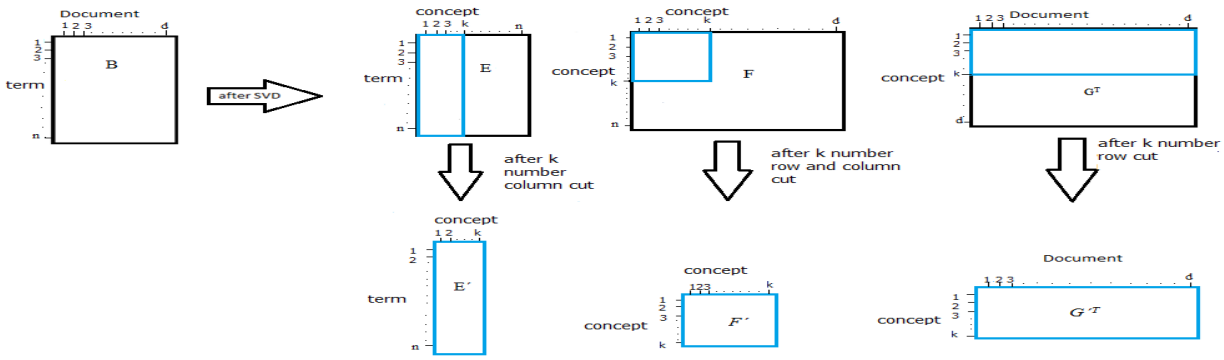

Figure1. SVD of term-by-document matrix

Latent Semantic Analysis (LSA) is an algebraic-statistical technique for representing meanings of words by their contextual usages and mapping documents into low dimensional abstract concept space where a concept is represented by the set of words appearing in the similar contextual usages [4]. In other words, it maps relations among terms and documents in semantic space. The rationale is that terms which occur in similar context will be positioned nearer to each other in the latent semantic space. The degree of relevance between documents and queries are then estimated by computing the cosine measure in the latent semantic space [21][22].

For LSI based document retrieval model, document representation is done in two steps. The first step is the creation of a term-by-document matrix, $B = [B_1, B_2, \ldots, B_d]$, where each column B_i represents the vector of term weights for the i-th document in the corpus. Here the frequency of a term is calculated as the number of times the term occurs in the document [23]. When we have computed term-document matrix of order $n \times d$, where n is the number of distinct terms in a document and d is the number of documents in the corpus. We have removed Bengali stop-word. After creating the term-by-document matrix, the next step is to apply singular value decomposition (SVD) which is applied to reduce the dimension and construct the latent semantic space. When SVD of the matrix B is performed, this results in three matrices M, N and P as follows:

$$B_{n\times d} = E_{n\times n}\, F_{n\times d}\, G^T{}_{d\times d} \qquad (6)$$

From NLP (Natural Language processing) point of view, E is term-by-concept matrix of size $n \times n$. F is concept-by-concept rectangular diagonal matrix of size n × d with positive real numbers in descending order on the diagonal. G^T is a concept-by-document matrix with size d × d. When the dimensionality of the matrices E, F and G^T are reduced to k most important dimensions, we can obtain three matrices: E´ is $n \times k$ *matrix*, F' is a $k \times k$ *matrix* and G'^T is $k \times d$ matrix. We demonstrate this in Figure1.

Since SVD maps n dimensional vectors (where a vector corresponds to a document) to k dimensional space by breaking down the original matrix B into k independent base vectors expressing k different concepts or topics in the corpus. A left singular vector which corresponds to a column of the matrix E represents a word combination pattern recurring in the corpus and one of the left singular vectors represents the most salient pattern. As each particular word combination pattern describes a certain concept in the corpus, the facts described above naturally lead to the hypothesis that each singular vector represents a salient concept of the corpus, and the magnitude of its corresponding singular value present in the matrix F represents the degree of importance of the salient concept. Similarly, each row of the matrix G^T represents a salient concept and an entry p_{ij} in the matrix G^T represents degree of similarity of the document j with the salient concept i.

1) Query Representation using LSI

For comparing queries with documents, they should be mapped to the same semantic space. For this purpose, the query vector is obtained by TF based representation similar to document representation. Then the query vector is projected into the k-dimensional subspace, and we denote the vector of query (q) as q_n. Then the new query vector in the reduced k-dimensional space is obtained as:

$$q_k = q_n^T \mathrm{E}'(F')^{-1} \qquad (7)$$

2) Ranking documents

The relevance of a document to a query is measured by the cosine similarity score, S, between the query vector q_k and each document vector, that is, a column of G^T corresponding to the document. For example, the relevance of document j to the query is computed as:

$$S_j = \frac{q_{k\,(G'^T)}}{||q_k||\,||G'^T||} \qquad (8)$$

Where $(G^{\prime T})_j$ is the j^{th} column of $G^{\prime T}$.

3) The Okapi BM25 model

Okapi BM25 [15] [16] is a one of the probabilistic retrieval model. The score is calculated following as:

$$BM25Score(d,q) = \frac{TF}{TF + (K1 * (1-B)) + B * (DLen/AvgDLen)} \times IDF \quad (9)$$

This score function is similar as TFIDF score function because in this equation have a TF part and IDF part[17]. In the above equation d is set as document and q is set as query. For long document have large TF values which are dominating effect on a document's score. So, normalize TF, a parameter K1 is used, to change TF to $\frac{TF}{TF+K1}$. The parameter B is used to control document lengths, resulting in the score of the BM25 model. BM25, as implemented by the Lemur Project, assigns a score to each document given a query term by the following score function. [16]

$$LemurBM25Score(d,q) = \frac{K3+1}{K3} \times IDF \times TFfactor \quad (10)$$

Where

$$TFfactor = \frac{(1+K1) \times TF}{TF + \big(K1 * (1-B)\big) + B \times (DocLen/AvgDocLen)} \quad (11)$$

Here K1 is used to control term frequency. K3 is used to give an extra weight to the entire score of document. B is used to control the document length normalization.

IV. Evaluation and Results

We have also implemented Traditional vector space model, Okapi BM25, LSI model for comparing models in Bengali Language. The retrieval engine has been tested for nine queries to search relevant documents from a corpus of approximately 3255 documents. The retrieval engine performance is measured by the terms of Mean Average Precision (MAP).

Average precision (AP) is calculated as follows.

$$AP(q_j) = \frac{1}{n_j} \sum P(tt)$$

- Here tt is the position of a relevant document in the ranked list, and P(tt) is the precision at position t.

Precision at position tt is calculated as follows.

$$P(tt) = \frac{rel\_tt}{tt}$$

Where rel_tt is the number of relevant documents retrieved till the position tt. MAP is calculated by computing the average of AP over all queries.

$$MAP(Q) = \frac{1}{|Q|} \sum AP(q_j) = \frac{1}{|Q|} \sum \frac{1}{n_j} \sum P(tt)$$

We used MAP to evaluate a retrieval model. Table 1 includes the average precision scores obtained by various IR models for our designed 9 different queries. We have compared three IR models for proving effectiveness of IR model. Model A is the TFIDF based traditional vector space model, model B is the LSI based IR model presented in this paper and model C is the Okapi BM25 IR model presented in this paper. We observed from the conducted experiments that BM25 model (Model C) generates the best MAP among the evaluated models for Bengali Language.

Table 1. Performance Comparisons of our developed three models based on MAP

IR Models	MAP (Mean Average Precision)
Model C	0.538
Model B	0.5078
Model A	0.5003

We have implemented model A, model B and model C after stemming queries and documents both with stop words removed and punctuation removed.

One of the important parameters in the model B is the value of k which indicates dimension of semantic space documents and queries are mapped to. K value is set from 10 to 3000. The model B gives best results when k is set to 95.
The BM25 score has 3 parameters, K1, B and K3, which need to be tuned for obtaining the better retrieval performance.
Typically, K1 is set approx. 0.2 to 3.0. The parameter B was set 0 to 1. It was observed that for high values of BM25 usually give the better performance for high value of K3. So K3 value changed repeatedly 50, 100, 150, 200, and 300. For BM25, the best MAP values obtained are shown. For our experiment, BM25 model with K1 set to 2.5, B set to 0.6, and K3 set to 250 gave the best score of 0.538.

V. Conclusion and Future Scope

In this study, we developed three IR models for Bengali language. The experimental results reveal that BM25 is more suitable among the three models for the Bengali Information Retrieval tasks. The current system does not use semantic

matching between query and documents. So, it can further be extended for concept based matching that means how a document is conceptually similar to the query. We have planned to apply deep learning based technique for computing conceptual matching between document and query.

ACKNOWLEDGMENT

This research work has received support from "Swami Vivekananda Merit Cum Means Scholarship" funded by Higher Education, Science and Technology and Bio-Technology Department, government of West Bengal.

REFERENCES

[1] R. Banerjee, & S. Pal, "ISM @ FIRE - 2011: Monolingual Task", In Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2011). Available at http://www.isical.ac.in/~fire/2011/workingnotes. html (visited May 2015),2011.

[2] U. Barman, P. Lohar, P. Bhaskar, & S. Bandyopadhyay, " Ad-hoc Information Retrieval focused on Wikipedia based Query Expansion and Entropy Based Ranking" ,Working Notes of the Forum for Information Retrieval Evaluation, Available at http://www.isical.ac.in/~fire/2012/working-notes.html, 2012.

[3] P. Bhaskar, Das, A. Pakra & S. Bandyopadhyay , "Theme Based English and Bengali Ad-hoc Monolingual Information Retrieval in FIRE 2010", In Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2010), Available at http://www.isical.ac.in/~fire/2010/working_notes.html (visited May 2015), 2010.

[4] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, & R. Harshman, "Indexing by latent semantic analysis", Journal of the American society for information science, Vol. 41, No. (6), 391. 1990.

[5] L. Dolamic & J. Savoy, "UniNE at FIRE 2008: Hindi, Bengali, and Marathi IR" , In: Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2008). Available at http://www.isical.ac.in/~fire/2008/working_notes.html (visited May 2015) ,2008.

[6] D. Ganguly, J. Leveling, & G. J. F. Jones, "A Case Study in Decompounding for Bengali Information Retrieval. Information Access Evaluation, Multilinguality, Multimodality, and Visualization, Lecture Notes in Computer Science, Vol. 8138, pp. 108-119,2013.

[7] M. Kantrowitz, B. Mohit, & V. Mittal ,"Stemming and Its Effects on TFIDF Ranking" In Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Athens, Greece ,pages 357–359, 2000.

[8] W. Kraaij & R. Pohlmann, "Viewing stemming as recall enhancement" In Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval, ACM ,pp. 40-48,1996.

[9] P. J. Loponen, , & K. Jarvelin, "UTA Stemming and Lemmatization Experiments in the Bengali ad hoc Track at FIRE 2010", In Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2010). Available at http://www.isical.ac.in/~fire/2010/working_notes.html (visited May 2015), 2010.

[10] P. Majumdar, M. Mitra,S.K. Parui & G. Kole, "YASS: Yet Another Suffix Stripper", ACM Transactions on Information Systems, Vol. 25 , No.4, Article 18,2007.

[11] R. Marcus, "Computer and Human Understanding in Intelligent Retrieval Assistance", American Society for Information Science, 28, 1998.

[12] P. McNamee, "N-gram Tokenization for Indian Language Text Retrieval", In Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2008), Available at http://www.isical.ac.in/~fire/2008/working_notes.html (visited May 2015), 2008.

[13] J. H. Paik & S. K. Parui, "A Simple Stemmer for Inflectional Language" , In Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2008), Available at http://www.isical.ac.in/~fire/2008/working_notes.html (visited May 2015), 2008.

[14] S. .E. Robertson, "The probability ranking principle in IR", Journal of Documentation, 33, 294-304, 1977.

[15] G. Salton, A. Wong & C. S. Yang, "A vector space model for automatic indexing" , Communications of the ACM, Vol.18, No.11, PP.613-620, 1975.

[16] K. Sarkar & A. Gupta, "An Empirical Study of Some Selected IR Models for Bengali Monolingual Information Retrieval", In Proceedings of ICBIM, NIT, Durgapur, 2016.

[17] K. Jones Spärck , S. Walker & S. E. Robertson, "A probabilistic model of information retrieval Development and comparative experiments", IP&M, Vol. 36, No. 6, pp.779–808, 809–840.

[18] H. Turtle & W. Bruce Croft, "Inference networks for document retrieval", InProc. SIGIR, pp. 1–24, 1989

[19] H. Turtle & W. Bruce Croft, "Evaluation of an inference network-based Retrieval model", TOIS ,Vol.9, No. 3, pp.187–222, 1991.

[20] C. J. Van Rijsbergen, "Information Retrieval", 2nd edition, Butterworths, LONDON, 1979.

[21] A. Singhal and F. Pereira, "Document expansion for speech retrieval" , In procedding of ACM SIGIR, Berkeley, CA, USA, pages 223-232,1999.

[22] M. Berry, S. Dumais and G. W. O'Brien, "Using linear algebra for intelligent information retrieval, SIAM Review, pp.573-595, 1995.

[23] D.R Radev, H. Jing, M. Sty´s, and D. Tam. Centroid-based summarization of multiple documents. Information Processing and Management,Vol. 40, No. 6,pp.919–938, 2004.

[24] S. Chatterjee & K. Sarkar, Combining "IR Models for Bengali Information Retrieval", International Journal of Information Retrieval Research (IJIRR), vol.8 issue 3 article 5, pp.68-83, 2017.

Authors Profile

Mrs. Soma Chatterjee completed her Master of Engineering from Jadavpur University in the year 2016. She is currently pursuing Ph.D. and full time Research Fellow in Department of Computational Sciences and Engineering, at Jadavpur University since 2017. She has published 2 research papers in international journals and 1 conference available online. Her main research work focuses on Clustering Based Information Retrieval and Summarization, NLP and Machine Learning based education.

Prof K. Sarkar received his B.E degree in Computer Science and Engineering from the Faculty of Engineering, Jadavpur University in 1996. He received the M.E degree and Ph.D. (Engg) in Computer Science and Engg. from the same University. In 2001, he joined as a lecturer in the Department of Computer Science & Engineering, Jadavpur University, Kolkata, where he is currently a professor. His research interest includes Natural Language Processing, Machine Learning, Text Summarization, Text Mining, Speech Recognition.

