# JADAVPUR UNIVERSITY

MASTER DEGREE THESIS

---

# BENGALI DOCUMENT INFORMATION RETRIEVAL USING QUERY EXPANSION

---

A thesis submitted in partial fulfillment of the requirements for the degree of

**Master of Technology in Computer Technology**
in the
**Department of Computer Science and Engineering**

By

**Srijan Patra**

University Roll Number : **001910504010**
Examination Roll Number : **M6TCT22012**
Registration Number : **149845** of **2019-20**

Under the Guidance of
**Prof. (Dr.) Kamal Sarkar**
Department of Computer Science and Engineering
Jadavpur University
Kolkata-700032

2021-2022

# FACULTY OF ENGINEERING AND TECHNOLOGY
# JADAVPUR UNIVERSITY

## Certificate of Recommendation

This is to certify that the thesis entitled "**BENGALI DOCUMENT INFORMATION RETRIEVAL USING QUERY EXPANSION**" has been carried out by **Srijan Patra** (University Roll No: **001910504010**, Examination Roll No: **M6TCT22012**, University Reg No: **149845** of 2019-20), under the guidance and supervision of Prof. (Dr.) **Kamal Sarkar**, Department of Computer Science and Technology, Jadavpur University, Kolkata, is being presented for partial fulfillment of the Degree of Master of Technology in Computer Technology during the academic year 2021-2022. The research results presented in the thesis have not been included in any other paper submitted for the award of any degree in any other university or institute.

Prof. (Dr.) **Kamal Sarkar** (Thesis Supervisor)
Department of Computer Science and Engineering
Jadavpur University, Ko1kata-32

Countersigned

Prof. (Dr.) **Anupam Sinha**
**Head of Department** ,
Computer Science and Engineering
Jadavpur University, Kolkata-32.

Prof. **Chandan Mazumdar**
**Dean** ,
Faculty of Engineering and Technology
Jadavpur University, Kolkata-32.

# FACULTY OF ENGINEERING AND TECHNOLOGY
# JADAVPUR UNIVERSITY

## Certificate of Approval

(Only in case the thesis is approved)

This is to certify that the thesis entitled "**BENGALI DOCUMENT INFORMATION RETRIEVAL USING QUERY EXPANSION**" is a bona fide record of work carried out by Srijan Patra (University Roll No: **001910504010**, Examination Roll No: **M6TCT22012**, University Reg No:**149845** of 2019-20) in partial fulfillment of the requirement for the Degree of Master of Technology in Computer Technology from the Department of Computer Science and Engineering, Jadavpur University during the period of August 2021 to July 2022. It is understood that by this approval the undersigned does not necessarily endorse or approve any statement made, opinion expressed, or conclusion drawn therein but approves the thesis only for the purpose for which it has been submitted.

Signature of Examiner 1

Date:

Signature of Examiner 2

Date:

# FACULTY OF ENGINEERING AND TECHNOLOGY
# JADAVPUR UNIVERSITY

## Declaration of Originality and Compliance of Academic Ethics

I hereby declare that this thesis entitled "**BENGALI DOCUMENT INFORMATION RETRIEVAL USING QUERY EXPANSION**" contains a literature survey and original research work by the undersigned candidate, as part of his degree of Master of Technology in Computer Technology.

All information in this document has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name (Block Letters): **SRIJAN PATRA**

Roll Number: **001910504010**
Examination Roll Number: **M6TCT22012**
University Registration Number: **149845 of 2019-20**

Thesis Title: **BENGALI DOCUMENT INFORMATION RETRIEVAL USING QUERY EXPANSION**

Signature with Date:

# ACKNOWLEDGMENT

I would like to express my deepest gratitude and sincere thanks to my respected mentor and teacher Prof . ( Dr. ) Kamal Sarkar , Department of Computer Science and Engineering , Jadavpur University for his exclusive guidance and undivided support in completing and presenting the thesis successfully. I am indebted to him for the constant encouragement and continuous knowledge sharing that he has given to me . The words above are a token of my deepest gratitude towards him for everything he has done to forge my thesis into what it is today.

I would also like to thank Ph.D. Research Scholar Soma Chatterjee , Department of Computer Science and Engineering , Jadavpur University for her unparalleled guidance and knowledge-sharing sessions which she provided during the complete duration of this thesis , without whom completing this thesis would not have been possible . I am thankful to her for her assistance and advice at every stage of the project.

In addition, I'd like to thank my classmates and the Department of Computer Science and Engineering at Jadavpur University for their assistance.

I am also grateful to my teachers in the Department of Computer Science and Engineering at Jadavpur University who have provided me with key insights over the years.

Finally, I want to thank my family and friends for their continual encouragement and support during my studies.

Regards ,

**SRIJAN PATRA**

Roll Number: **001910504010**
Examination Roll Number: **M6TCT22012**
University Registration Number: **149845 of 2019-20**
Department of Computer Science and Engineering
Jadavpur University

Signature with Date :

# CONTENTS

**LIST OF FIGURES**

## LIST OF TABLES

## LIST OF ACRONYMS

**BM :** Best Matching
**CBOW :** Continuous Bag of Words Model
**CLEF :** Conference and Labs of the Evaluation Forum
**FAQ :** Frequently Asked Questions
**FIRE :** Forum for Information Retrieval Evaluation
**IDF :** Inverse Document Frequency
**IF :** Information Filtering
**IL :** Indian Language
**IR :** Information Retrieval
**LOD :** Linked Open Data
**MAP :** Mean Average Precision
**MIT :** Massachusetts Institute of Technology
**ML :** Maximum Likelihood
**NLTK :** Natural language Toolkit
**NTCIR :** National Center for Science Information Systems
**QA :** Question Answering
**QALD :** Question Answering over Linked Open Data
**QE :** Query Expansion
**CLIR :** Cross-Language
**NLP :** Natural Language Processing
**TF :** Term Frequency
**TREC :** Text Retrieval Conference

# ABSTRACT

The goal of Information Retrieval (IR) is to retrieve documents from a vast document collection whose content fits a user query. Since most users struggle to create well-designed queries, query expansion is required to extract relevant information. Query expansion (QE) is an IR process that includes selecting and adding terms to a user's query in order to decrease query document mismatch and hence increase retrieval performance. Various QE techniques are commonly used to boost the efficiency of textual information retrieval systems. The amount of Indian language (IL) electronic documents has increased significantly as a result of globalization. As a result, the need for developing IR systems to deal with this growing collection is paramount. In this thesis , we propose a Bengali document Information Retrieval system using Query Expansion which implements the Vector Space Paradigm of Information Retrieval and uses the Pseudo Relevance Feedback approach of Query Expansion. Using our Information Retrieval System, we create Initial Search Results for a certain query. We re-analyze the top documents from the Initial Search Results and reformulate the initial query using our term selection algorithm. To obtain the Final Search Results, we take the reformulated query as the expanded query and rerun it through the same Information Retrieval System. For our experiments we have proposed seven comparison models. All experiments have been performed on Queries 100 to 125 of the FIRE 2010 dataset. analyzing the MAP of these said seven models against the MAP of the baseline model. Experimental results show that the MAP of our proposed method using QE improves over the MAP of the baseline Information Retrieval System without using QE . The hybrid collaboration model outperforms the baseline model on a higher level. The MAP of this model on the final search results is significantly higher than the MAP of our baseline model on initial search results without employing QE, demonstrating that the Pseudo Relevant Feedback approach of QE does actually aid in the retrieval of more relevant Bengali documents.

# 1. INTRODUCTION

Information retrieval is a growing topic that addresses a wide range of issues related to the storage and retrieval of diverse media. Our major focus is on text document storage and retrieval in response to a user's information request [1].

In information retrieval, a document is any unit of text that has been indexed and is retrievable in the system. Depending on the context, a document can represent anything from common things like newspaper articles or encyclopedia entries to tiny components like excerpts and sentences.

Users can access a vast array of reference materials using information retrieval (IR) technologies. The issue is dealing with this massive quantity of information in determining how to properly offer relevant information to the user in response to a query [2] .

A collection is a set of documents used to fulfill user requests. A term is a lexical entity found in a collection, however it can also include phrases. Finally, a query is a combination of phrases that conveys a user's information need[3]. The drive of a person or an alliance to find and acquire knowledge to fulfill a conscious or unconscious need is referred to as information need.

As a result, information retrieval is defined as the act of getting information resources appropriate to a user's information demand from a collection of information resources.

In an ideal monolingual IR system, information on a subject supplied by the user should be retrieved in natural language. This retrieval process entails a user interaction with the IR system, in which the user enters a query and the system returns a collection of relevant documents. The user query will be a natural language text with semantic ambiguity. To be accurate, the IR system should interpret the document collection based on the user query and rank them based on their relevancy. Both syntactic and semantic information from the document collection should be inferred in order to retrieve the exact content connected to the query.

Massive development in the field of information retrieval (IR) has been documented during the previous fifty years, yet most of it has been done in English . Aside from that, a number of IR communities (such as TREC, CLEF, NTCIR, and FIRE) have initiated notable projects in a variety of East Asian, European, and South Asian languages.

This thesis focuses on Information Retrieval of Bengali documents on the datasets provided by FIRE .

A Bengali monolingual Information Retrieval application accepts queries in Bengali and must return a sorted list of Bengali documents based on their relevance to the query.

Our objective is not only to create a Bengali monolingual Information Retrieval application that ranks Bengali documents based on their relevance but also to improve the precision of the ranked results of the said Bengali monolingual Information Retrieval application by using query expansion .

Query expansion (QE) is a procedure in information retrieval that consists of choosing and adding words to the user's query with the purpose of decreasing query document mismatch and so enhancing retrieval performance.

The primary concept is to reformulate the query using the results of the initial search and then execute a second run of search to acquire results with greater accuracy .

## 1.1 Information Retrieval Process

In Information Retrieval, the query process is composed of two main phases, indexing and matching . It is also possible to broaden the searches in order to increase retrieval performance.

The indexing stage prepares documents and queries for use in queries by obtaining keywords (relevant words, also known as terms). At this stage, stemming and stop word lists should be considered in order to reduce linked terms to their stem, base, or root form.

This is accomplished by initiating affix removal, which converts distinct derivational or inflectional forms of the same word to a single indexing form and removes words that do not carry information important to the content.

Matching is the process of determining the similarity of documents and queries by weighting phrases, with the TF-IDF and BM25 algorithms being the most commonly used. In response to a query, most retrieval systems produce a ranked document list, with the documents most similar to the query assessed by the system appearing first on the list.

After obtaining the initial set of answers, various query expansion strategies may be used. We employ the pseudo relevance feedback method.

In order to evaluate the results of the retrieval process, MAP or Mean Average Precision is used . It summarizes rankings from multiple queries by calculating the mean of average precision .

## 1.2 Query Expansion

Query expansion(QE). is the activity of reformulating a query to improve retrieval efficiency in information retrieval operations .

It is a computer science methodology that has been addressed in the areas of natural language processing and information retrieval [4].

Query expansion (QE) is an information retrieval process that includes selecting and adding terms to a user's query in order to decrease query document mismatch and hence increase retrieval performance. It reformulates the user's original query to increase information retrieval efficacy[5].

Typically there are four phases of query expansion: (i) data source preprocessing and term extraction, (ii) term weights and ranking, (iii) term selection, and (iv) query reformulation (see Fig. 1)

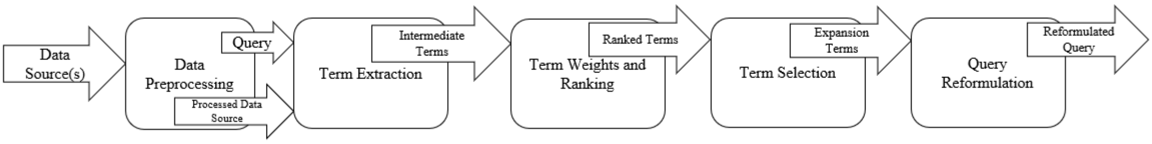


Fig. 1 : Query Expansion Process

## 1.2.1 Phases of Query Expansion

### 1.2.1.1 Data Source Preprocessing and Term Extraction

Preprocessing a data source is determined by the data source and the QE technique employed; it is not determined by the user's query. This step's main purpose is to extract a collection of words from the data source that adds value to the user's initial query. It comprises the four steps listed below[5]:

1. Text extraction from the data source (extraction of whole texts from the specific data source used for QE)
2. Tokenization (the process of splitting the stream of texts into words)
3. Stop word removal (removal of frequently used words, e.g., articles, adjectives, prepositions, etc.)
4. Word stemming (the process of reducing derived or inflected words to their base word)

### 1.2.1.2 Term Weights and Ranking

The mechanism utilized to give term weights in the document and query vectors has a significant impact on the retrieval system's efficacy. Two elements have been shown to be crucial in the development of efficient term weights. The first is term frequency, which is just the raw frequency of a term inside a document [6]. This component represents the assumption that terms that appear often within a text may more strongly reflect its meaning than terms that appear less frequently and should thus have greater weights[1].

The second component is used to give terms that appear only in a few documents more weight. Terms that are restricted to a few documents are beneficial for distinguishing those papers from the rest of the collection; terms that appear often across the collection aren't as valuable[1].

### 1.2.1.3 Term Selection

It is possible that the selected QE approach generates a high number of expansion terms, but it is not always feasible to employ all of these expansion terms. Typically, only a small number of expansion words are chosen for QE. This is related to noise reduction, which occurs when a query with a small collection of expansion words outperforms a query with a large set of expansion terms [7].

### 1.2.1.4 Query Reformulation

This is the final phase in the QE process, in which the expanded query is rebuilt to get better results when used to retrieve relevant documents. The reformulation is done based on the weights assigned to the individual terms of the expanded query; this is known as query reweighting [5].

## 1.2.2 Importance of Query Expansion

One of the most important benefits of QE is that it increases the likelihood of retrieving useful information from any document collection that would not be recovered otherwise using the original query[5]. Many times, the user's original query is insufficient to get the information that the user intends or seeks. In this circumstance, QE is extremely important.

Numerous QE approaches may be employed to increase retrieval precision, each with its own set of advantages. A few core approaches of QE are shown in Fig. 2.

Recent strategies have been described as being based on either global or local analysis of the documents in the corpus being searched[5].

The global approaches look at word occurrences and associations in the corpus as a whole and utilize this knowledge to broaden any specific query [8] .

Local analysis, on the other hand, only includes the top-ranked documents returned by the original query. It is named local since the approaches are extensions of the initial work on local feedback [8].

Both global and local analysis have the advantage of expanding the query. Inherently, the global analysis is more expensive than the local analysis[8].

These approaches of Query Expansion are described in proper with proper examples in the Section 2 of this thesis.

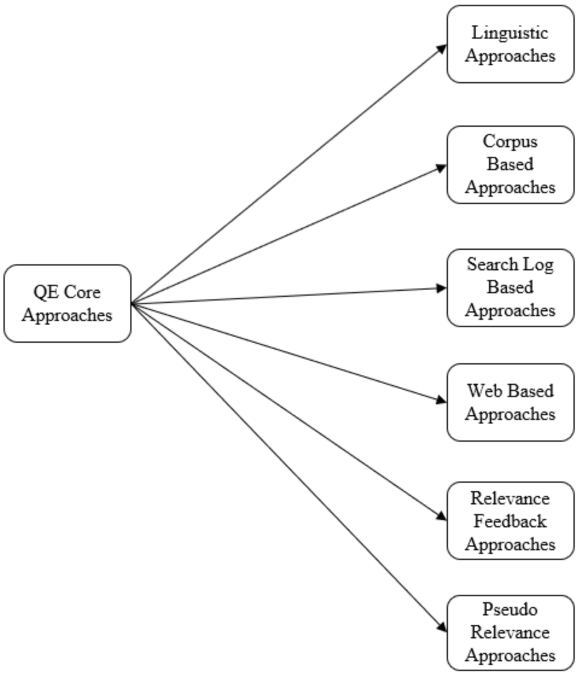

QE Core Approaches
Linguistic Approaches
Corpus Based Approaches
Search Log Based Approaches
Web Based Approaches
Relevance Feedback Approaches
Pseudo Relevance Approaches

Fig. 2 : Core Approaches Of QE

## 1.2.3 Applications of Query Expansion

Aside from the primary field of IR, recently there have additional applications where QE approaches have proven useful. The various approaches of Query Expansion are namely personalized social documents , Question Answering , CLIR , Information Filtering , Multimedia IR , Indian Language IR , etc.

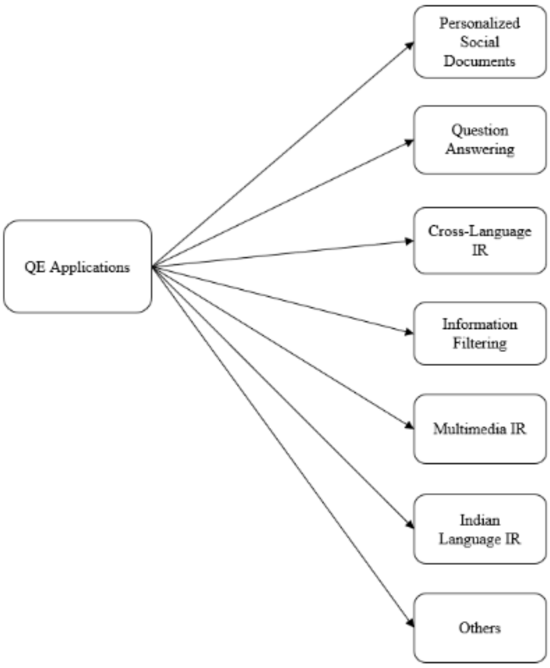


Fig. 3 : Applications Of QE

### 1.2.3.1 Personalized social documents

Social tagging systems have gained popularity in recent years due to their usage in the sharing, labelling, commenting, rating, and other aspects of multi-media material. Every user wants to find information that is relevant to his or her interests and obligations. This has resulted in the need for a QE framework based on social bookmarking and tagging systems, which improve document representation.

Bender [9] proposes a QE framework for leveraging the many elements available in social interactions (people, documents, tags) and their mutual relationships. It also computes score functions for each entity and its relationships. For tailored online searches, Biancalana and Micarelli [10] employ social tagging and bookmarking in QE. Their testing results reveal that the user's interests are effectively matched with the search results.

Bouadjenek [9] uses a mix of social proximity and semantic similarity for tailored social QE, which is based on similar phrases that a certain user and his social relatives frequently use.

Zhou [11] presented a QE approach based on unique user profiles, in which the expansion words are retrieved from both the annotations and the resources provided and selected by the user.

### 1.2.3.2 Question Answering

Question answering (QA) has emerged as a significant study subject in the realm of IR systems. The basic goal of QA is to provide a rapid response to the user's inquiry. The goal here is to keep the answer succinct rather than to get all relevant papers.

Recently, search engines have begun to use the QA system to deliver responses to such inquiries. However, one of the key obstacles in QE for evaluating the responses to such questions is the mismatch problem, which emerges owing to a mismatch between the expression in question and the text-based answers [12].

Much research has been conducted in order to eliminate the mismatch problem and enhance document retrieval in QA systems. Agichtein, Lawrence, and Gravano proposed an interesting method for QE utilizing FAQ data [13] .

Riezler, Vasserman, Tsochantaridis, Mittal, and Liu [14] describe a method for expanding the user's initial inquiry in a QA system by utilizing Statistical Machine Translation (SMT) to bridge the lexical gap between questions and responses. SMT seeks to bridge the language gap between the user's query and the system's response. This system's purpose is to learn the lexical connections between words and phrases in questions and answers[5].

Wu [15] extend short questions by mining user intent from three separate sources, namely the CQA archive, query logs, and online search results. QE in Question Answering over Linked Open Data (QALD) is now gaining popularity in the field of natural language processing.

Shekarpour, Höffner, Lehmann, and Auer [16] suggested a strategy for expanding the initial query on linked data utilizing linguistic and semantic characteristics collected from WordNet and semantic features retrieved from the Linked Open Data cloud.

The experimental findings reveal that the accuracy and recall rates are significantly higher than in the baseline techniques.

### 1.2.3.3 Cross-Language IR

Cross-Language IR or CLIR is a subset of IR that obtains information in languages other than the user's query language. For example, a user can inquire in Hindi, yet the essential information returned may be in English.

Traditional CLIR research issues include untranslatable query words, phrase translation, inflected terms, and ambiguity in language translation between the source and destination languages [17].A typical method for overcoming this translation issue is to employ QE [18].

It produces a better result – even when there is no translation problem – since statistical semantic similarity among the terms is used [19]. Gaillard, Bouraoui, De Neef, and Boualem [20] employ linguistic resources for QE to compensate for faults in automated machine translation in cross-language inquiries. QE can be used at different stages of the translation process, including before, after, and both.

It has been demonstrated that the application prior to translation produces better results than the application post-translation; nevertheless, the application at any level produces better results than not employing QE [21] [22] [23] [24].

### 1.2.3.4 Information Filtering

Information filtering (IF) is a technique for removing non-essential information from a dataset and delivering just the relevant results to the end-user. Information filtering is widely utilized in many sectors, including Internet search, e-mail, e-commerce, multimedia distributed systems, blogs, and so on. IF may be divided into two categories: content-based filtering and collaborative filtering.

Several QE techniques have been reported to improve the relevance of data produced following IF. Relevance feedback strategies broaden the user's query in ways that match the user's interests and requirements [25]. To improve results, Eichstaedt [26] combines the user's query with the system's master query.Wu, Liu, Xie, Ester, and Yang [27] improved the Collaborative Filtering strategy by reformulating the query by utilizing the user-item co-clustering method [5] .

### 1.2.3.5 Multimedia IR

Multimedia Information Retrieval or MIR is concerned with searching for and extracting semantic information from multimedia documents such as audio, video, and picture. Most MIR systems rely on text-based searches for IR in multimedia materials, such as titles, captions, anchor text, annotations, and surrounding HTML or XML representation. When metadata is missing or cannot accurately represent the actual multimedia material, this strategy may fail. As a result, QE is critical in obtaining the most important multimedia data.

A popular strategy for finding spoken audio is to do a text search on the transcription of the audio file. However, because the transcription is generated automatically by voice translation software, it has inaccuracies. In such a case, expanding the transcription by adding related words greatly improves the retrieval effectiveness [28] . According to Jourlin, Johnson, Jones, and Woodland [29], QE can enhance average precision in audio retrieval by 17%.

Queries and documents in video retrieval involve both visual and textual aspects. The expanded text searches are matched with the text descriptions of the graphic conceptions that have been manually created.

Natsev, Haubold, Tei, Xie, and Yan [30] expand text query for visual QE by using lexical, statistical, and content-based techniques.

A frequent way of retrieving relevant photos in image retrieval involves searching by utilizing textures, forms, color, and visual aspects that match the image descriptions in the database.

Kuo, Chen, Chiang, and Hsu [31] provide two QE approaches: intra expansion (expanded query is acquired from existing query features) and inter expansion (expanded query is obtained from the search results).

Hua [32] looks for generic online images using query log data.

Xie, Zhang, Tan, Guo, and Li [33] provide a contextual QE approach for overcoming the semantic gap in visual vocabulary quantization, as well as the performance and storage loss caused by QE in image retrieval.

### 1.2.3.6 Indian Language IR

Indian languages belong to the Indo-European class of languages. Hindi and Bengali are two of the world's top seven most widely spoken languages. The number of Indian language (IL) electronic papers has increased significantly in recent years. As a result, the need for developing IR systems to deal with this growing collection is irrefutable.

A number of IR communities (like TREC, CLEF, NTCIR and FIRE) have taken noticeable initiatives in several East-Asian, European and South-Asian languages.

Many IR systems are already being created for languages such as Hindi, Bengali, Malayalam, and Urdu utilizing basic IR models. Only a few researchers make use of Query Expansion to boost their results.

Researchers have already started to use Query Expansion to get information in languages such as Malayalam ,Urdu and Bengali [34][35][36]. These works are though in their preliminary stages in the world of QE.

The structure of these languages is not as simple as compared to English , so , the IR systems of these languages can always be bettered especially using different query expansion approaches .

This thesis deals with Bengali Information Retrieval using the local analysis QE approach of Pseudo Relevance Feedback on the datasets provided by FIRE. The primary concept is to reformulate the query using the results of the initial search and then execute a second run of search to acquire greater accuracy results .

### 1.2.3.7 Others

Plagiarism detection [37], event search [38] [39], text classification [40], patent retrieval [41], dynamic process in IoT [42], e-commerce classification, biomedical IR [43], enterprise search [44], code search [45], parallel computing in IR [46], and twitter search [47] are some other recent applications of QE.

## 1.3 Organization of Thesis

The description of the rest of the thesis is mentioned here.

**Section 2** describes a brief survey of the literature on Information Retrieval and Query Expansion. **Section 3** describes the dataset that has been used. **Section 4** describes our methodology for Bengali document IR using Query Expansion. **Section 5** describes the evaluation metrics , experiments used and results that are obtained. **Section 6** describes the implementation of the project for the thesis. **Section 7** draws the conclusion of the proposed work and states the directions for future work.

# 2. LITERATURE SURVEY

Information retrieval is a developing discipline that covers a wide variety of problems connected to the storage and retrieval of various media. Our primary focus is on the storing of text documents and their subsequent retrieval in response to a user's information request [1] .

A document in information retrieval refers to any unit of text that has been indexed in the system and is retrievable. A document can represent anything from common objects like newspaper articles or encyclopedic entries to smaller components like excerpts and phrases, depending on the use.

A collection is a group of documents that are used to attain user requests. A term is a lexical entity that appears in a collection ; however , it can also comprise phrases. Finally, a query expresses a user's information need as a set of terms[3].

An individual's or an alliance's urge to discover and receive information to meet a conscious or unconscious need is known as information need.

Thus, information retrieval is the process of acquiring information resources relevant to a user's information need across a collection of information resources.

## 2.1 Paradigms of Information Retrieval

A model is a representation of a real-world process that is idealized or abstracted. The computing process may be described using information retrieval models.

Information retrieval models can be categorized into three paradigms :

- Boolean Retrieval Model
- Vector Space Model
- Language Model

### 2.1.1 Boolean Retrieval Model

The Boolean retrieval model is an information retrieval paradigm in which each query may be expressed as a Boolean expression of words, that is, terms coupled with the operations AND, OR, and NOT. Each document is seen as a collection of words by the model [3].

It is a traditional model of information retrieval that was the earliest and most widely used. Almost all commercial IR systems currently utilize it.

The documents are indexed in advance , hence it is possible to avoid scanning the contents in a linear fashion for each query. Suppose we record for each document – for example a play of Shakespeare's – whether it contains each word out of all the words Shakespeare used . A binary term-document incidence matrix emerges as a consequence [3]. The indexed units are called terms. We may now have a vector for each term that displays the documents it occurs in, or a vector for each document that shows the terms that occur in it, depending on whether we examine the matrix rows or columns.

Assume a binary term-document incidence matrix with a size of 500K * 1M , having half-trillion 0s and 1s, which is too large to fit in a computer's memory. The important point is that this matrix is highly sparse, meaning it has few non-zero entries. For example, if each document is 1000 words long, the matrix contains no more than one billion 1's, implying that 99.8% of the cells are zero. Recording only the things that do exist, i.e. the 1 positions, is a far better portrayal. The inverted index, the first significant notion in information retrieval, is based on this principle[3].

Despite decades of academic study on the benefits of ranked retrieval, large commercial information providers used systems based on the Boolean retrieval model as their primary or only search option for three decades, until the early 1990s. These systems, on the other hand, didn't only feature the fundamental Boolean operations (AND, OR, and NOT) that we've seen so far. For many of the information needs that people have, a strict Boolean expression over terms with an unordered results set is insufficient, hence these systems built extended Boolean retrieval models that included extra operators such as term proximity operators[3].

Marcus invented the Smart Boolean [48] . It attempts to assist users in the construction and modification of a Boolean inquiry, as well as in making better choices along the many dimensions that describe a Boolean query.

Boolean query models are preferred by many users, particularly professionals. Boolean inquiries are precise: a document either matches or does not match the query. This gives the user more control and transparency over the information that is obtained. Furthermore, some areas, such as legal documents, allow for successful document ranking inside a Boolean paradigm. This does not, however, imply that Boolean searches are more successful for expert searchers. A typical issue with Boolean search is that employing AND operators produces high accuracy but low recall searches, while using OR operators produces low precision but high recall searches, and finding an acceptable medium ground is difficult or impossible [2] .

## 2.1.2 Vector Space Model

In the vector space model of information retrieval , documents and queries are represented as vectors of features representing the term that occur within the collection [1] .

The value of each feature is called the term weight and is usually a function of the term's frequency in the document , along with other factors . Terms are axes of space and documents are points or vectors in this space [1].

Vector space representation [49] is a distinct document representation that is used to categorize documents and is utilized in information retrieval systems. Each document is regarded as a vector in this context. Because terms are high-dimensional axis, they are normalized to vectors of one length.

We employ the cosine metric instead of the real angle in vector-based information retrieval. The cosine of the angle between two vectors is used to calculate the distance between two documents. When two texts are identical, the cosine is 1; when they are orthogonal (have no common terms), the cosine is 0. As a result, another way to make sense of cosine is as the normalized dot product, which is the dot product of the two vectors divided by the lengths of each vector [1].

All of the basic components for an ad hoc retrieval system are provided by this categorization of documents and queries as vectors. A document retrieval system can simply receive a user's query, convert it to a vector, compare it to the vectors representing all known documents, and sort the results. The result is a list of documents ranked according to how closely they match the query[1].

The ability to see the document collection as a sparse matrix of weights is enabled by the portrayal of documents as vectors of term weights, where $w_{i,j}$ denotes the weight of term i in document j [1].

A term-by-document matrix is the most generic term for this weight matrix. The columns of the matrix in this view reflect the documents in the collection, while the rows represent the words.

### 2.1.3 Language Model

Thinking about terms that would likely exist in a relevant document and using those words as the query is a typical tip to users for coming up with appropriate searches. This principle is clearly modelled in the language modelling method to IR: a document is a good match to a query if the document model is likely to create the question, which will happen if the document includes the query terms frequently. As a result, this technique offers a unique interpretation of some of the fundamental concepts in document ranking.

Ponte and Croft pioneered the language modelling technique [50]. A novel method of document scoring was developed. It is referred to as the query likelihood score. It was suggested that a document be considered a bag of words and that a document can yield a query. If a document can create a query, then it is considered to be relevant to the inquiry.

The query likelihood may be estimated using two types of probabilities, according to this model: (1) the chance that a query word present in the document is created by the document, and (2) the probability that a query word absent in the document is generated by the document.

One issue with this maximum likelihood (ML) estimator is that an unknown word in document D would have a zero probability, resulting in a zero probability for all queries including an unseen word, which is obviously undesired.

More significantly, when a document is relatively little, the ML estimate is usually incorrect. So, one essential difficulty to overcome is smoothing the ML estimator so that we don't assign zero probability to unseen words which may increase the overall accuracy of the estimated language model[2].

Rather than explicitly modelling the probability P (R = 1|q, d) of a document d's relevance to a query q, as in the traditional probabilistic approach to IR, the basic language modelling approach creates a probabilistic language model $M_d$ from each document d and ranks documents based on the probability of the model generating the query : P (q|$M_d$) [3].

The concept of a language model is probabilistic by definition. A language model is a function that calculates the likelihood of strings from a vocabulary. We model the query probability given the document rather than the document probability given the query.

The most basic language model just ignores any conditioning context and estimates each term alone. A model like this is known as a unigram language model[3]:

There are a variety of more complicated language models, such as bigram language models, which are dependent on the prior word[3].

Language models based on grammar, such as probabilistic context-free grammars, are significantly more sophisticated. However, unigram language models have been employed in the majority of IR language modelling studies because IR does not directly rely on the structure of words to the extent that other procedures like speech recognition do. Unigram models are frequently sufficient for determining a text's topic[3].

Language modelling is a broad formal approach to IR that has a variety of implementations. The query likelihood model is the earliest and most basic way of employing language models in IR. We create a language model $M_d$ from each document d in the collection. Our goal is to rank documents using P(d | q), where P(d | q) refers to a document's probability of being relevant to the query.

## 2.2 Query Expansion

The practice of reformulating a query to increase retrieval performance in information retrieval activities is known as query expansion (QE). It is a computer science methodology that has been addressed in the areas of natural language processing and information retrieval [4].

Query expansion (QE) is a procedure in information retrieval that involves choosing and adding phrases to a user's query in order to reduce query document mismatch and hence improve retrieval performance. It reformulates the user's original query to improve the efficacy of information retrieval[5].

Rocchio [51] established relevance feedback in the vector-space paradigm in 1965. Sparck Jones [52] and van Rijsbergen [53] pioneered the use of collection-based word co-occurrence statistics to identify query expansion terms.

According to Krovetz and Croft [54] , expanded set T, which is derived based on term similarity boosts the recall rate in query results. As a result, the selection of set T and the selection of data sources D are important components of QE research.

A document is similarly indexed by the natural language words that make up its content or by a set of regulated index terms that map its contents to ideas in a specified domain. Both the process of directly matching free-text query keywords to free-text index terms and the process of converting natural language words to restricted vocabularies are inherently imperfect. The key issue in the first scenario is that the user and the creator of a document may convey the same notion using a different language.

The shades of meaning that natural language phrases hold may be lost in the translation process in the second situation. In addition to these issues, the user's query may be partial or wrong, i.e., the user may not articulate or express his/her information requirement precisely or properly.

The goal of query expansion is to enrich the user's query by finding additional search terms that represent the user's information need more accurately and completely, avoiding, at least to some extent, the aforementioned problems and increasing the chances of matching the user's query to representations of relevant ideas in documents.

QE approaches may be categorized as follows in terms of automation and end-user engagement[5] :

- **Manual Query Expansion :** The user must manually reformulate the query in this case.[55]
- **Automatic Query Expansion :** The system reformulates the query without the need for human participation in this case. The system's intelligence includes both the method for computing set T′ and the choice of data sources D.[55]
- **Interactive Query Expansion :** In this case, query reformulation occurs as a result of the system and the user working together. It's a human-in-the-loop strategy in which the system offers search results based on an automatically reformulated question, and users choose the most relevant ones. The system reformulates the query and obtains the results based on the user's preferences. The procedure is repeated until the user is content with the search results. .[55]

Researchers agree that adding selected terms enhances retrieval performance, the estimated ideal number ranges from a few terms to a few hundred terms. Various people have different ideas on how many selected terms should be added: one-third of the original query terms [56], five to ten terms [57] [58], 20–30 terms [59], 30–40 terms [60], a few hundreds of terms [61], and 350–530 terms for each query [62].

These terms may have been derived from the most frequently retrieved documents or from well-known relevant documents. It has been discovered that adding these expansion words boosts retrieval efficacy from 7% to 25% [62]. On the contrary, several research indicates that the number of terms used for QE is less relevant than the terms chosen based on type and quality [63]. It has been well demonstrated that the efficiency of QE declines minutely as the number of non-optimal expansion terms increases [64].

The majority of experimental research found that the number of expansion terms is irrelevant and fluctuates from query to query [65]. When fewer than 20 expansion terms are included, the efficiency of QE (measured as mean average accuracy) drops [60] [40]. In most cases, 20–40 terms are the ideal choice for QE.

Salton and Buckley [66] introduced a popular query reweighting approach that is influenced by Rocchio's method [67] for relevance feedback and its subsequent advancements.

## 2.2.1 Classification of Query Expansion Approaches

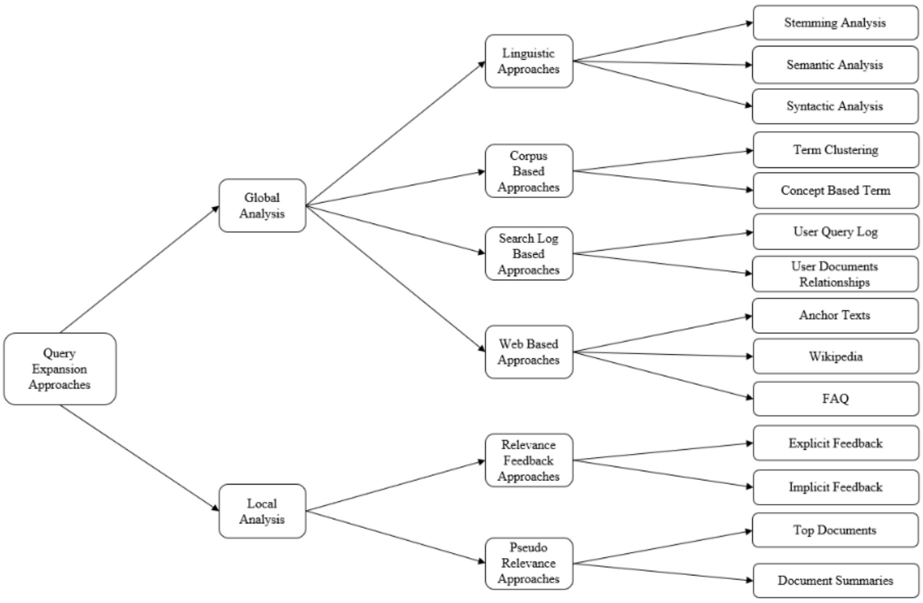


Fig. 4 : Query Expansion Approaches

Several ways have been offered based on the data sources used in QE. All of these techniques are divided into two categories: (1) global analysis and (2) local analysis. As illustrated in Fig. 4, global and local analyses may be further subdivided into four and two subclasses, respectively. This section explores QE techniques depending on the characteristics of various data sources utilized in QE as seen in Fig 4.

### 2.2.1.1 Global Analysis

In the global analysis, QE approaches implicitly choose expansion words for reformulating the initial query from hand-built knowledge resources or big corpora. For broadening the initial inquiry, only individual query terms are examined. The expanded words have semantic similarities to the original ones. Each term is given a weight; the expansion terms might be given less weight than the original query terms. On the basis of query terms and data sources, the global analysis may be divided into four categories: (i) linguistic-based, (ii) corpus-based, (iii) search log-based, and (iv) web-based. Each strategy[5] is briefly explained in the sections that follow:

#### 2.2.1.1.1 Linguistic Based Approaches

To reformulate or extend the initial query terms, the methodologies in this category examine expansion aspects such as lexical, morphological, semantic, and syntactic term associations. They make use of thesauruses, dictionaries, ontologies, the Linked Open Data (LOD) cloud, and other knowledge resources like WordNet or ConceptNet.

Word stemming is one of the first and most prominent QE techniques in linguistic association for reducing inflected words to their base words. The stemming algorithm [68] can be used either during retrieval or during indexing.

During retrieval, phrases from originally obtained texts are selected and then harmonized with the morphological kinds of query terms [69].

Other common QE techniques in the linguistic association include semantic and contextual analysis. Ontologies, LOD clouds, dictionaries, and thesaurus are examples of knowledge sources. Bhogal [70] employs domain-specific and domain-independent ontologies in the context of ontologically based QE. Wu, Ilyas, and Weddell [71] use domain ontology's rich semantics to assess the trade-off between improved retrieval efficacy and computational cost. Several studies on QE have been conducted utilizing a thesaurus. WordNet is a well-known thesaurus that may be used to broaden the initial query by employing word synsets. As previously stated, many research studies employ WordNet to broaden the initial query.

Syntactic analysis [72] is another key strategy for improving the linguistic information of the first question. Syntactic QE expands the initial inquiry by utilizing the increased relational properties of the query words. It typically widens the query by statistical methodologies [71].

#### 2.2.1.1.2 Corpus Based Approaches

Corpus-based techniques analyze the whole text corpus to identify the expansion characteristics to be used for QE.

They were among the first statistical methods for QE. They employ co-occurrence data in the corpus to build phrases, paragraphs, or surrounding words, which are then used in the enlarged query. Corpus-based techniques can use one of two strategies: (1) term clustering [73], which divides document words into clusters based on their co-occurrences, or (2) concept based terms [74], which base expansion terms on the concept of the query rather than the original query terms. Kuzi [75] chooses the expansion terms after analyzing the corpus via word embeddings, in which each phrase in the corpus is represented by an embedded vector.

#### 2.2.1.1.3 Search Log Based Approaches

These methods are based on the examination of search records. The study of search logs is commonly used to investigate user input, which is an essential source for proposing a collection of related phrases depending on the user's initial query. With the rapid expansion of the internet and the increased usage of online search engines, the volume of search logs and their simplicity of use has made them a key source of QE. It typically comprises user queries that correlate to Web page URLs. Cui [76] extracts probabilistic correlations between query phrases and document terms from query logs. These correlations are then utilized to broaden the user's original inquiry.

On the basis of online search logs, two types of QE techniques are typically deployed. The first kind treats inquiries as documents and extracts query characteristics relevant to the user's original inquiry [77].

In the second technique, characteristics are retrieved based on the relational behavior of queries. For example, Baeza-Yates and Tiberi [78] encode requests in a graph-based vector space model (query-click bipartite graph) and analyze the resulting graph using query logs.

#### 2.2.1.1.4 Web Based Approaches

These methods include using Wikipedia and anchor texts from websites to broaden the user's initial inquiry. These techniques have recently acquired favor. McBryan [79] was the first to employ anchor text to associate hyper-links with connected pages as well as pages where anchor texts are available. In the context of a web page, anchor text can serve a similar function to the title in that it can act as a brief summary of the page's contents.

Another common method is to leverage Wikipedia articles, titles, and hyper-links (in- and out-links) [80]. As we all know, Wikipedia is the largest free online encyclopedia; articles are continually updated, and new ones are published on a daily basis. These characteristics make it an excellent information source for QE.

FAQs are another valuable web-based resource for enhancing the QE. Karan & Šnajder [81] released a paper in which they employ domain-specific FAQs data for manual QE.

### 2.2.1.2 Local Analysis

Local analysis techniques include QE approaches that choose expansion terms from the collection of documents returned in response to the user's first (unmodified) query. The working assumption is that the documents obtained in response to the user's original inquiry are relevant, thus words found in these documents should be relevant to the initial question as well. There are two approaches to broaden the user's initial query using local analysis: (1) Relevance feedback and (2) Pseudo-relevance feedback. These will be described further below[5].

#### 2.2.1.2.1 Relevance Feedback

The user's input regarding documents retrieved in response to the original inquiry is gathered in this technique; the feedback is about whether the retrieved documents are relevant to the user's query. The query is recast depending on the documents identified as relevant by the user. Rocchio's technique [67] was one of the first to use relevance feedback. There are two sorts of relevance feedback: explicit feedback and implicit feedback.

The user actively assesses the relevance of obtained articles in explicit feedback [66], but in implicit feedback, the user's activity on the set of documents retrieved in response to the original query is utilized to implicitly deduce the user's preferences [82] [83]. The lack of semantics in the corpus hinders relevance feedback [71].

#### 2.2.1.2.2 Pseudo Relevance Feedback

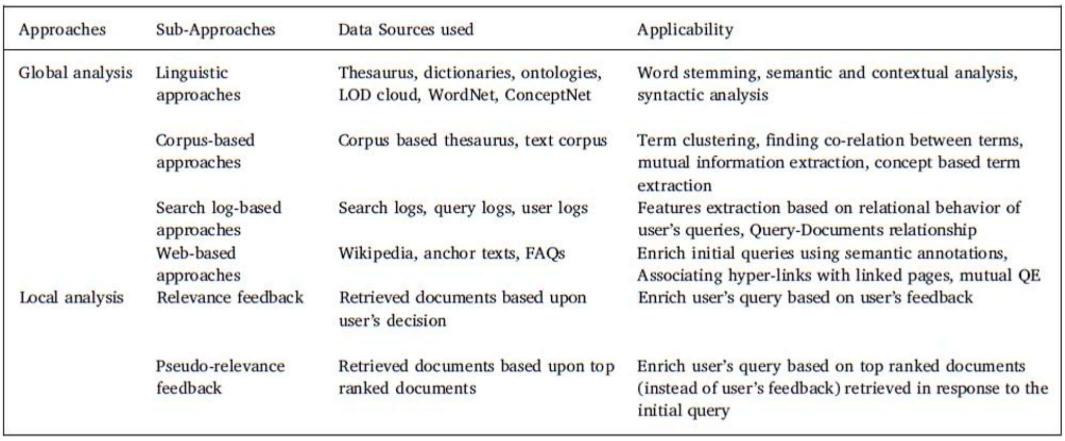
The user's explicit or implicit input is not gathered here. Instead, the feedback gathering procedure is automated by employing the top-ranked documents (or their excerpts) returned in response to the original query straight for QE.

Blind feedback, or retrieval feedback, is another name for pseudo-relevance feedback.

Croft and Harper initially presented this strategy in 1979 [84], and they use it in a probabilistic model. Xu and Croft [85] suggested an approach called "local context analysis" to extract QE terms from the leading documents returned in response to the first query. The co-occurrence of query terms is used to award a score to each of the possible expansion terms. For query reformulation, the candidate terms with the maximum ranking are chosen.

Several ways have been presented in addition to using the top-ranked documents or their excerpts. Techniques based on passage extraction [86], text summarization [87], and document summaries [88] are examples.

In conclusion, there is a large range of QE techniques with distinct features that are generally effective or suitable in certain contexts. The optimal solution is determined by weighing numerous criteria, such as the kind of queries, the availability and characteristics of external data sources, the type of collection being searched, the facilities provided by the underlying weighting and ranking system, and the efficiency requirements.



| Approaches | Sub-Approaches | Data Sources used | Applicability |
|---|---|---|---|
| Global analysis | Linguistic approaches | Thesaurus, dictionaries, ontologies, LOD cloud, WordNet, ConceptNet | Word stemming, semantic and contextual analysis, syntactic analysis |
| | Corpus-based approaches | Corpus based thesaurus, text corpus | Term clustering, finding co-relation between terms, mutual information extraction, concept based term extraction |
| | Search log-based approaches | Search logs, query logs, user logs | Features extraction based on relational behavior of user's queries, Query-Documents relationship |
| | Web-based approaches | Wikipedia, anchor texts, FAQs | Enrich initial queries using semantic annotations, Associating hyper-links with linked pages, mutual QE |
| Local analysis | Relevance feedback | Retrieved documents based upon user's decision | Enrich user's query based on user's feedback |
| | Pseudo-relevance feedback | Retrieved documents based upon top ranked documents | Enrich user's query based on top ranked documents (instead of user's feedback) retrieved in response to the initial query |

Table 1 : Applicability of QE techniques categorized with respect to data sources.

## 2.2.2 Query Expansion for Indian Languages

The Indian subcontinent can be regarded as another Europe, due to its lingual diversity. This region of the world has a total population of roughly 1,900 million people who speak approximately 25 official languages.

Among the primary languages of this area, Hindi and Bengali are among the world's top 7 most spoken languages . Sanskrit and the Dravidian languages are the foundations of most common Indian languages.  Over the last few years, there has been a significant increase in the number of Indian language (IL) electronic documents.

As a result, the requirement for building IR systems to deal with this expanding repository is unquestionable.

A number of IR communities (like TREC, CLEF, NTCIR and FIRE) have taken noticeable initiatives in several East-Asian, European and South-Asian languages.

Many IR systems based on fundamental IR models are already being developed for languages such as Hindi, Bengali, Malayalam, and Urdu. Query Expansion is used by just a few researchers to improve their results.

Early IR research was primarily concerned with the creation of IR techniques for English. Recently, there has been an increase in interest in the creation and automatic assessment of information retrieval systems for Indian languages. Sarkar and Gupta [89] give a comparative analysis of the performance of several information retrieval methods in Bengali.

Dolamic and Savoy [90] and Paik and Parui [91] provided various IR models for Bengali, Hindi, and Marathi languages.

Banerjee and Pal[92], Bhaskar [93], Ganguly[94], Loponen, and Paik [95] have all described ways of Bengali monolingual retrieval.

Researchers have undertaken a number of attempts to construct IR systems for Indian languages, with the majority of their efforts focusing on enhancing the stemming process of the classic vector space model for IR systems [2] .

Ganguly [96] strays from this trend to some extent by studying the effect of de-compounding for Bengali IR.

Query Expansion has previously been used by researchers to obtain information in languages such as Malayalam, Urdu, and Bengali [34][35][36]. In the realm of QE, these efforts are still in their early phases.

The structure of these languages is not as straightforward as that of English, their IR systems may constantly be improved, particularly by adopting alternative query expansion methodologies.

Barman [97] used Wikipedia for query expansion and Entropy-based ranking. For Hindi – English cross-lingual IR, Chandra and Dwivedi [98] presented a method for query extension based on term selection.

Using the distributed neural language model word2vec, Roy, Paul, Mitra, and Garain [99] provided a framework for Automatic Query Expansion (AQE).The goal here is to locate words that are semantically linked to a particular user query and leverage word embeddings in order to improve QE effectiveness [99] .

Sarkar and Maity [100] suggested a retrieval method for Bengali tweets based on sentiment analysis employing the Twitter dataset. Sentiment analysis is the method of extracting sentiment from text using natural language processing (NLP).

Islam, Rahman Talha, and Chowdhury[34] published a paper in which they offered a query expansion technique for selecting expanded terms that were be used with a search key to improve the accuracy and efficiency of search results for a Bengali Search Engine , Pipilika[34]

This thesis deals with Bengali Information Retrieval using the local analysis QE approach of Pseudo Relevance Feedback on the 2010 datasets provided by FIRE . The primary concept is to reformulate the query using the results of the initial search and then execute a second run of search to acquire greater accuracy results .

# 3. DATASET DESCRIPTION

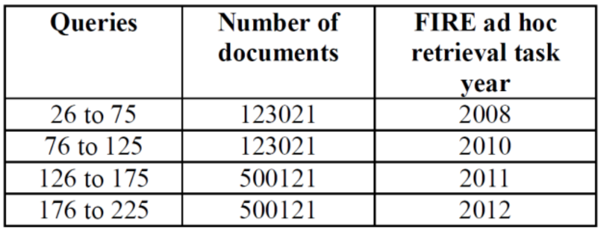
FIRE was founded in 2008 to create a South Asian counterpart to TREC, CLEF, and NTCIR, and has subsequently grown to meet the new problems in multilingual information access.

For ad hoc retrieval assignments, the Forum for Information Retrieval Evaluation (FIRE) prepared a dataset of Bengali papers. There are 500122 documents in the collection.

It includes news stories from reputable newspapers such as Anandabazar Patrika and BDNews. It covers items from Anadabazar Patrika dating from 2001 to 2010. It covers items from BDNews dating from 2006 through 2010.

All files are UTF-8 encoded. We have used the FIRE ad hoc task 2010 dataset. There are a total of 123021 documents in the collection.

Table 2 shows the queries used in various FIRE ad hoc retrieval tasks, as well as the number of documents in the corpus that year [89] .



| Queries | Number of documents | FIRE ad hoc retrieval task year |
|---|---|---|
| 26 to 75 | 123021 | 2008 |
| 76 to 125 | 123021 | 2010 |
| 126 to 175 | 500121 | 2011 |
| 176 to 225 | 500121 | 2012 |

Table 2 : Number of Bengali Queries in Retrieval tasks for different years

For our experiments and evaluation we have used Queries 100 to 125 on an indexed collection of 713 documents .

For the first round :

**Input:** Queries 100 to 125
**Output:** Initial Search results

For the second round :

**Input:** Queries 100 to 125, Initial Search results
**Output:** Final Search Results

# 4. METHODOLOGY

## 4.1 Proposed Method

We have proposed a hybrid approach to Query Expansion for Bengali Information Retrieval which implements the Vector Space Paradigm of Information Retrieval and uses the Pseudo Relevance Feedback approach of Query Expansion .

For a particular query we generate Initial Search Results using our Information Retrieval System . We then re-analyze Top N1 Documents from the Initial Search Results and use our term selection algorithm to reformulate the initial query. We use the reformulated query as the expanded query and run it on the same Information Retrieval System to provide us with the Final Search Results. The working of the model has been illustrated in Fig 5.

## 4.2 Proposed Method Architecture

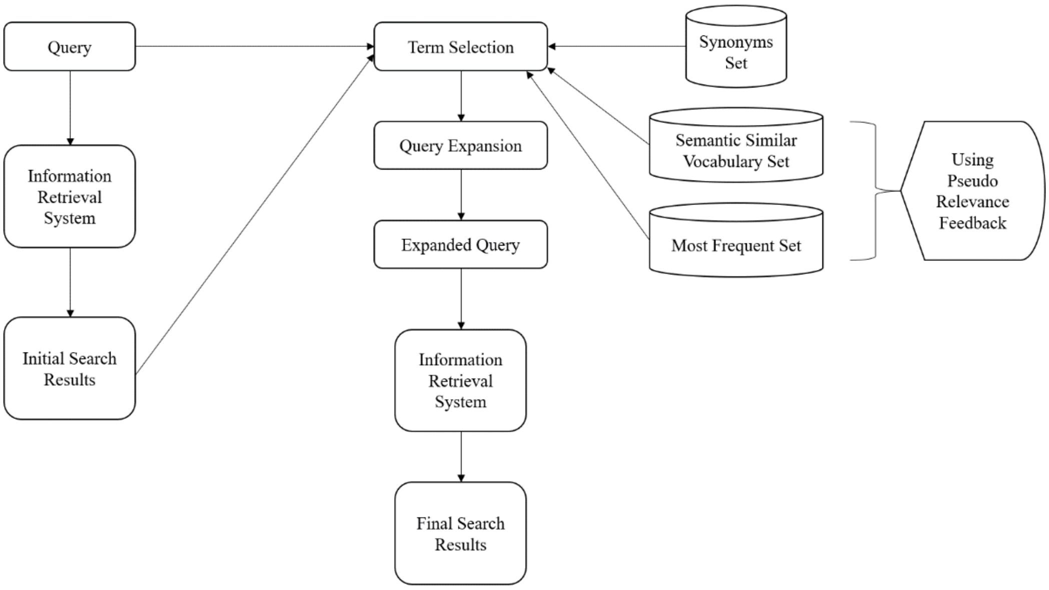


Fig. 5 : Proposed Method Architecture

## 4.3 Generic Framework for Bengali IR using BM25

The Information Retrieval System used implements the Vector Space paradigm of Information Retrieval and ranks relevant documents using the Okapi BM25 scoring function.

### 4.3.1 Information Retrieval System Architecture

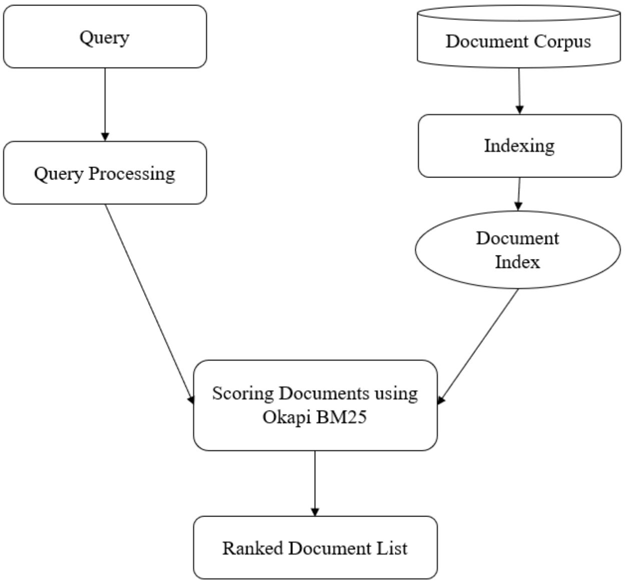


Fig. 6 : Information Retrieval System Architecture

### 4.3.2 Indexing and Query Processing

Tokenization, stemming/stop word removal, and storing the information on file using a particular data structure for rapid access during query processing are the three key phases in indexing.

All documents had their first punctuation deleted. Stop-words were then eliminated from the text using FIRE's stop-word list.

Using the Bag-of-words concept, the documents are tokenized into a collection of words. Following that, stemming was performed utilizing the stem list supplied by FIRE.

When a user enters a query into the system, it is sent to the retrieval system. The retrieval system forwards the query to the query processing phase, which handles the query in the same manner as the document. The query is also subjected to tokenization, stop word removal, and stemming. Following the first processing of the query, it is given to a vector space model for ranking, which matches a query vector to the document vectors.

Let a user query consist of n terms Q = $\{t_1, t_2, ..., t_i, t_{i+1}, ..., t_{n+1}\}$.

The reformulated query can include two parts :

new terms T′=$\{ t'_1, t'_2,..., t'_m \}$ from the data source(s) D, and stop words T″=$\{ t''_{i+1}, t''_{i+2},..., t''_n \}$ from the data source(s) D [5]. The revised query looks like this:

$$Q_{exp} = ( Q - T'' ) \cup T' = \{t_1, t_2, ..., t_i, t'_1, t'_2,..., t'_m \} \quad (1)$$

The set T′ is a collection of additional meaningful phrases added to the user's initial query in order to get more relevant content and eliminate ambiguity, as described above.

## 4.3.3 Okapi BM25

Okapi BM25, also known as BM25, is a weighting function that is used to rank documents based on their relevance to a particular query [101]. Many scholars use the BM25 tool to retrieve important documents from various corpora.

The following information may be derived from each document in a collection of documents C containing terms from a vocabulary V.

- **Term Frequency :** The Term Frequency of a Term measures the frequency of a word in a document. It is represented as $tf_{i,j}$ for a term i in document j .
  Using the log frequency weighting scheme the term frequency can be represented as :

$$log\ tf_{i,j} = \{ 1 + \log ( tf_{i,j}) \} \quad (2)$$

- **Document Frequency :** For a word, the DF measures the number of documents in the collection C, the term i is present. It is represented as $n_i$ .

- **Inverse Document Frequency :** It is the inverse of the DF. So, if a term is rare, it has a low DF and a high IDF, but if it appears in a significant number of papers in the collection, it has a high DF and a low IDF. The formula for calculating IDF is:

$$idf_i=\log( \frac{N}{n_i} ) \quad (3)$$

Some difficulties have been faced if the logarithm of TF and IDF is used . When TF is 0 , the calculation of a logarithm of 0 will be required . To avoid this situation , the logarithm of (1+TF) is calculated.

$$idf_i=\log( \frac{1+N}{1+ n_i} ) \quad (4)$$

BM25 is a probabilistic model that assigns weight to search terms based on their frequency inside the document and the frequency of the query term. The associated weighting function is as follows[102] :

**Okapi Scoring Function :**

$$BM25Sscore_{(d,q)} = IDF \; . \; \frac{(k_1+1) \; . \; TF}{k_1 \; . \; [\,(1-b) + b \; . \; (\, dl \,/\, avdl \,)\,] + TF} \; . \; \frac{(k_3+1)}{(\, k_3 \,)} \qquad (5)$$

(i) $\boldsymbol{k_1}$ , $\boldsymbol{b}$, and $\boldsymbol{k_3}$ are parameters that depend on the queries and the dataset;

(ii) $\boldsymbol{TF}$ is the occurrence frequency of the term in the document $d$;

(iii) $\boldsymbol{IDF}$ is the inverse document frequency ;

(iv) **dl** and **avdl** are, respectively, the document length and the average document length in the corpus;

(v) **d** is the document and **q** is the query.

Here, $\boldsymbol{k_1}$ is used to normalize$\boldsymbol{b}$ is used to normalize document lengths and $\boldsymbol{k_3}$ is used to add weight to the entire score . For our scoring purpose we set the values of $\boldsymbol{k_1}$ , $\boldsymbol{b}$, and $\boldsymbol{k_3}$ as $\boldsymbol{k_1 = 2.2}$ **,** $\boldsymbol{b}$**=0.3, and** $\boldsymbol{k_3 = 1}$ .

# 4.4 Term Selection

## 4.4.1 Proposed Term Selection Algorithm

**Input:** Query, Initial search results
**Output:** Selected terms for query expansion

**Step 1 :** Pre-processed Query Terms Pool= {Query Terms} (stop words removed)

Candidate Terms Pool = { empty set } ( initial )

**Step 2:** Extract the synonyms of the pre-processed query terms from a synonym vocabulary API , say S = extracted synonyms

Candidate Terms Pool = Candidate Terms Pool ∪ S

**Step 3:** Extract the terms from the semantic similar vocabulary set based on the similarity of each pre-processed query terms created in step 1 with the vocabulary terms.

The similarity between a query word and a vocabulary word is calculated using word vectors. Choose the vocabulary words whose similarity with the query word is > T (T =0.7 initially , and tuned).

The semantic similar vocabulary set is made up of document collection of top N1 documents from the initial search results. This based on Pseudo Relevance Feedback approach of QE .

Say, V= selected vocabulary terms.

$$\text{Candidate Terms Pool} = \text{Candidate Terms Pool} \cup V$$

**Step 4:** Choose the most frequent top K1 terms from the top ranked documents N2 from the initial search results . This is also based on Pseudo Relevance Feedback approach of QE .

- Say F= the chosen most frequent terms ( Top K1 words from top N2 documents )

$$\text{Candidate Terms Pool} = \text{Candidate Terms Pool} \cup F$$

**Step 5:** Ranking candidate terms

- Each term t in Candidate Terms Pool is checked for repeats and also stemmed to its root stem word , i.e. no two words now have the same root stem .
- The root stem word of each term t in Candidate Terms Pool is checked against each stem word of the pre-processed query terms . If the root stem word for a term t exists in the stemmed list of the pre-processed query terms , then the term t is discarded.
- Each term t in Candidate Terms Pool is assigned a weight which is computed as follows:

$$\mathbf{W = \alpha * context\ score + (1 - \alpha) * frequency\ score} \qquad (6)$$

where:

**context score** = Cosine between word vector of t and mean word vector for pre-processed query terms.

**frequency score** = frequency of term t in the initial search results returned by IR initially/MaxFreq ,

**MaxFreq** = maximum of the frequencies of the terms in top N2 ranked documents of the initial search results

**α** is a tuning parameter which controls the assigned weight and its value ranges from 0 to 1 .

- Ranks the candidate terms based on the values of **W**

**Step 6:** Return (top ranked K2 terms). K2 is specified by user and is also a tuning parameter.

## 4.4.2 Proposed Term Selection Algorithm Flow with examples

The proposed term selection algorithm is applied for Bengali Information Retrieval using Python 3.7 with a few modifications.

**Input:** Query, Initial search results
**Output:** Selected terms for query expansion

The detailed flow of the algorithm , along with examples for the Query 100 of the FIRE 2010 dataset are shown in the following subsections:

### 4.4.2.1 A Query Document

Fig. 7 shows a sample Query Document :

```
<title>অবৈধ পাসপোর্ট মামলায় মণিকা বেদী</title>
<desc>মণিকা বেদীর বিরুদ্ধে হায়দরাবাদ থেকে জাল পাসপোর্ট করানোর অভিযোগ</desc>
<narr>প্রাসঙ্গিক নথিতে মণিকার বিরুদ্ধে নাম ভাঁড়িয়ে হায়দরাবাদে পাসপোর্ট জালিয়াতির
অভিযোগ বা সে বিষয়ে সিবিআই তদন্ত সংক্রান্ত তথ্য থাকা চাই। অন্য কোথাও করানো জাল
পাসপোর্ট সংক্রান্ত তথ্য এখানে প্রাসঙ্গিক নয়।</narr>
```

Fig. 7 : A sample Query Document

### 4.4.2.2 Tokenized Query Terms

The terms in <title> , <desc> and <narr> tags are tokenized and grouped into a single list . Fig. 8 shows the tokenized query.

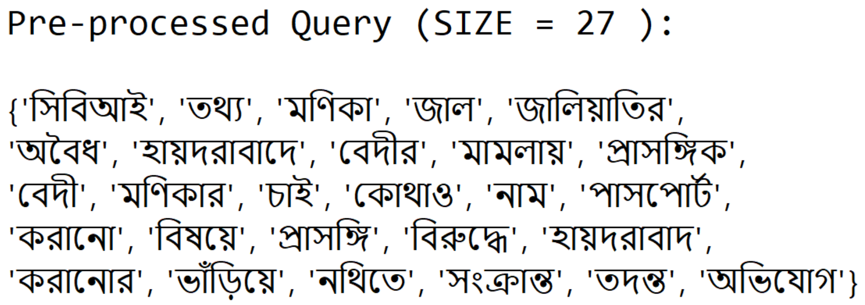
```
[' অবৈধ পাসপোর্ট মামলায় মণিকা বেদী মণিকা
বেদীর বিরুদ্ধে হায়দরাবাদ থেকে জাল পাসপোর্ট
করানোর অভিযোগ প্রাসঙ্গিক নথিতে মণিকার
বিরুদ্ধে নাম ভাঁড়িয়ে হায়দরাবাদে পাসপোর্ট
জালিয়াতির অভিযোগ বা সে বিষয়ে সিবিআই
তদন্ত সংক্রান্ত তথ্য থাকা চাই অন্য কোথাও করানো
জাল পাসপোর্ট সংক্রান্ত তথ্য এখানে প্রাসঙ্গিক নয় ']
```

Fig. 8 : Tokenized Query

### 4.4.2.3 Pre-processing of Query Terms

From this tokenized list , stop words and duplicates are removed . The stop words are removed with the help of stop word list provided by the Forum for Information Retrieval Evaluation (FIRE) for Bengali Information Retrieval . Fig. 9 shows the Query Terms after removing stop words . This pool is known as the Pre-processed Query Pool .

Pre-processed Query Terms Pool = {Query Terms} (stop words removed)



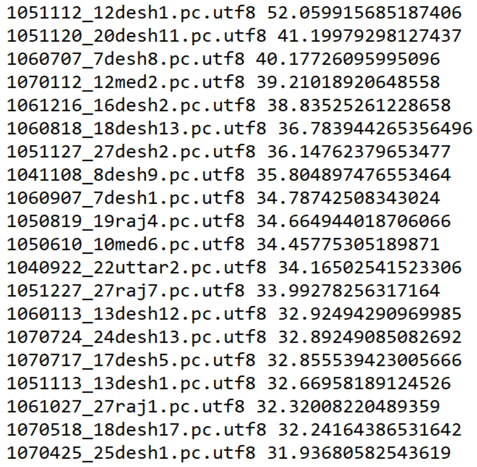
```
Pre-processed Query (SIZE = 27 ):

{'সিবিআই', 'তথ্য', 'মণিকা', 'জাল', 'জালিয়াতির',
'অবৈধ', 'হায়দরাবাদে', 'বেদীর', 'মামলায়', 'প্রাসঙ্গিক',
'বেদী', 'মণিকার', 'চাই', 'কোথাও', 'নাম', 'পাসপোর্ট',
'করানো', 'বিষয়ে', 'প্রাসঙ্গি', 'বিরুদ্ধে', 'হায়দরাবাদ',
'করানোর', 'ভাঁড়িয়ে', 'নথিতে', 'সংক্রান্ত', 'তদন্ত', 'অভিযোগ'}
```

Fig. 9 : Pre-processed Query Terms Pool removing stop words

### 4.4.2.4 Corresponding Initial Search Results

The Initial Search Results for this particular Query is calculated using our Information Retrieval System.

Our IR System provides us with a list of top 100 ranked documents along with its score using the Okapi BM25 ranking function . A snippet of the top 20 ranked documents is shown below in Fig.10:



```
1051112_12desh1.pc.utf8 52.059915685187406
1051120_20desh11.pc.utf8 41.19979298127437
1060707_7desh8.pc.utf8 40.17726095995096
1070112_12med2.pc.utf8 39.21018920648558
1061216_16desh2.pc.utf8 38.83525261228658
1060818_18desh13.pc.utf8 36.783944265356496
1051127_27desh2.pc.utf8 36.14762379653477
1041108_8desh9.pc.utf8 35.804897476553464
1060907_7desh1.pc.utf8 34.78742508343024
1050819_19raj4.pc.utf8 34.664944018706066
1050610_10med6.pc.utf8 34.45775305189871
1040922_22uttar2.pc.utf8 34.16502541523306
1051227_27raj7.pc.utf8 33.99278256317164
1060113_13desh12.pc.utf8 32.92494290969985
1070724_24desh13.pc.utf8 32.89249085082692
1070717_17desh5.pc.utf8 32.855539423005666
1051113_13desh1.pc.utf8 32.66958189124526
1061027_27raj1.pc.utf8 32.32008220489359
1070518_18desh17.pc.utf8 32.24164386531642
1070425_25desh1.pc.utf8 31.93680582543619
```

Fig. 10 : Top 20 ranked documents in Initial Search Results

### 4.4.2.5 Adding Synonyms

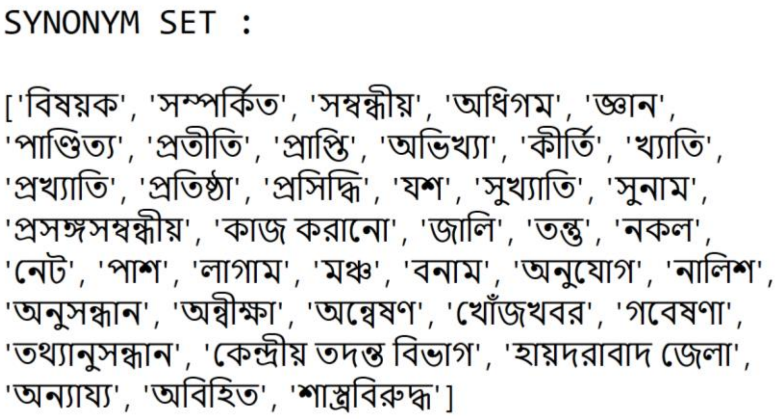
For each term in Pre-processed Query Terms Pool, synonyms are added using a python package named Bangla NLTK . This package has an MIT License and its author is Piyal Roy . It was released on Aug 4 , 2020.

This package provides us with appropriate synonyms.

Say S = extracted synonyms ,

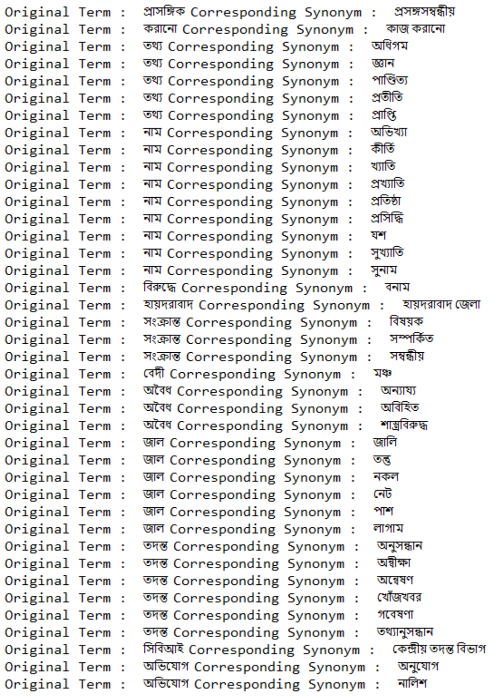
$$\text{Candidate Terms Pool} = \text{Candidate Terms} \cup S$$

Fig. 11 shows the selected synonyms set and Fig. 12 show all the original terms and their corresponding synonyms for the said Query.



```
SYNONYM SET :

['বিষয়ক', 'সম্পর্কিত', 'সম্বন্ধীয়', 'অধিগম', 'জ্ঞান',
'পাণ্ডিত্য', 'প্রতীতি', 'প্রাপ্তি', 'অভিখ্যা', 'কীর্তি', 'খ্যাতি',
'প্রখ্যাতি', 'প্রতিষ্ঠা', 'প্রসিদ্ধি', 'যশ', 'সুখ্যাতি', 'সুনাম',
'প্রসঙ্গসম্বন্ধীয়', 'কাজ করানো', 'জালি', 'তন্তু', 'নকল',
'নেট', 'পাশ', 'লাগাম', 'মঞ্চ', 'বনাম', 'অনুযোগ', 'নালিশ',
'অনুসন্ধান', 'অন্বীক্ষা', 'অন্বেষণ', 'খোঁজখবর', 'গবেষণা',
'তথ্যানুসন্ধান', 'কেন্দ্রীয় তদন্ত বিভাগ', 'হায়দরাবাদ জেলা',
'অন্যায্য', 'অবিহিত', 'শাস্ত্রবিরুদ্ধ']
```

Fig. 11 : Synonyms Set



```
Original Term :  প্রাসঙ্গিক Corresponding Synonym :  প্রসঙ্গসম্বন্ধীয়
Original Term :  করানো Corresponding Synonym :  কাজ করানো
Original Term :  তথ্য Corresponding Synonym :  অধিগম
Original Term :  তথ্য Corresponding Synonym :  জ্ঞান
Original Term :  তথ্য Corresponding Synonym :  পাণ্ডিত্য
Original Term :  তথ্য Corresponding Synonym :  প্রতীতি
Original Term :  তথ্য Corresponding Synonym :  প্রাপ্তি
Original Term :  নাম Corresponding Synonym :  অভিখ্যা
Original Term :  নাম Corresponding Synonym :  কীর্তি
Original Term :  নাম Corresponding Synonym :  খ্যাতি
Original Term :  নাম Corresponding Synonym :  প্রখ্যাতি
Original Term :  নাম Corresponding Synonym :  প্রতিষ্ঠা
Original Term :  নাম Corresponding Synonym :  প্রসিদ্ধি
Original Term :  নাম Corresponding Synonym :  যশ
Original Term :  নাম Corresponding Synonym :  সুখ্যাতি
Original Term :  নাম Corresponding Synonym :  সুনাম
Original Term :  বিরুদ্ধে Corresponding Synonym :  বনাম
Original Term :  হায়দরাবাদ Corresponding Synonym :  হায়দরাবাদ জেলা
Original Term :  সংক্রান্ত Corresponding Synonym :  বিষয়ক
Original Term :  সংক্রান্ত Corresponding Synonym :  সম্পর্কিত
Original Term :  সংক্রান্ত Corresponding Synonym :  সম্বন্ধীয়
Original Term :  বেদী Corresponding Synonym :  মঞ্চ
Original Term :  অবৈধ Corresponding Synonym :  অন্যায্য
Original Term :  অবৈধ Corresponding Synonym :  অবিহিত
Original Term :  অবৈধ Corresponding Synonym :  শাস্ত্রবিরুদ্ধ
Original Term :  জাল Corresponding Synonym :  জালি
Original Term :  জাল Corresponding Synonym :  তন্তু
Original Term :  জাল Corresponding Synonym :  নকল
Original Term :  জাল Corresponding Synonym :  নেট
Original Term :  জাল Corresponding Synonym :  পাশ
Original Term :  জাল Corresponding Synonym :  লাগাম
Original Term :  তদন্ত Corresponding Synonym :  অনুসন্ধান
Original Term :  তদন্ত Corresponding Synonym :  অন্বীক্ষা
Original Term :  তদন্ত Corresponding Synonym :  অন্বেষণ
Original Term :  তদন্ত Corresponding Synonym :  খোঁজখবর
Original Term :  তদন্ত Corresponding Synonym :  গবেষণা
Original Term :  তদন্ত Corresponding Synonym :  তথ্যানুসন্ধান
Original Term :  সিবিআই Corresponding Synonym :  কেন্দ্রীয় তদন্ত বিভাগ
Original Term :  অভিযোগ Corresponding Synonym :  অনুযোগ
Original Term :  অভিযোগ Corresponding Synonym :  নালিশ
```

Fig. 12 : Original Terms and their corresponding synonyms

### 4.4.2.6 Adding Semantic Similar Words using Word Embedding

Word embedding is a phrase used in natural language processing to describe the representation of words for text analysis, often in the form of a real-valued vector that encodes the meaning of the word such that words that are near in the vector space are considered to be similar in meaning.

For our semantic similar vocabulary set we choose the Top N1 documents from the Initial Search Results. Each distinct word in this Top N1 document collection is known as a vocabulary word. The similarity between a query word and a vocabulary word is calculated using word vectors.

We set the value of N1 as follows : **N1 =10**

The cosine similarity is calculated for each query word against each vocabulary word . We choose the vocabulary words whose cosine similarity with the query word is > T (T =0.7 ) .

For appropriate word vectors we use a python package known as FastText. It is regulated and maintained by Meta . FastText provides pre-trained word vectors for 157 languages, which have been learned using Common Crawl and Wikipedia. CBOW with position-weights in dimension 300, character n-grams of length 5, a window of size 5, and 10 negatives were used to train these models.

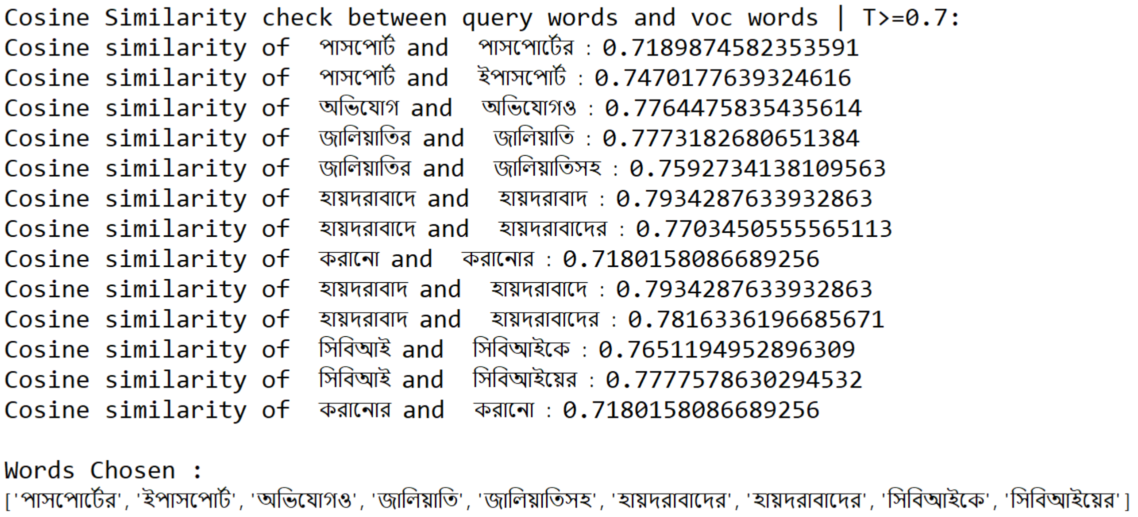
The purpose of using the previously specified threshold value of T >=0.7  in our scenario is to avoid introducing noise .

Fig. 13 shows the similar words that are chosen according to the cosine similarity metrics and the final chosen similar word list . Here N1=10 , i.e. the top 10 documents from the Initial Search Results are chosen as the document collection for our external vocabulary .

The chosen semantic similar words are added to the present Candidate Terms Pool .

Say, V= selected vocabulary terms.

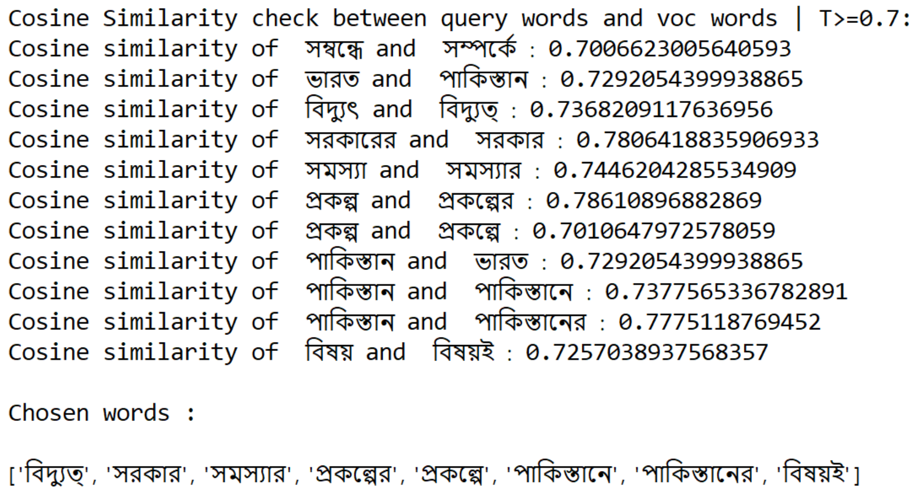
$$\text{Candidate Terms Pool} = \text{Candidate Terms Pool} \cup V$$



```
Cosine Similarity check between query words and voc words | T>=0.7:
Cosine similarity of  পাসপোর্ট and  পাসপোর্টের : 0.7189874582353591
Cosine similarity of  পাসপোর্ট and  ইপাসপোর্ট : 0.7470177639324616
Cosine similarity of  অভিযোগ and  অভিযোগও : 0.7764475835435614
Cosine similarity of  জালিয়াতির and  জালিয়াতি : 0.7773182680651384
Cosine similarity of  জালিয়াতির and  জালিয়াতিসহ : 0.7592734138109563
Cosine similarity of  হায়দরাবাদে and  হায়দরাবাদ : 0.7934287633932863
Cosine similarity of  হায়দরাবাদে and  হায়দরাবাদের : 0.7703450555565113
Cosine similarity of  করানো and  করানোর : 0.7180158086689256
Cosine similarity of  হায়দরাবাদ and  হায়দরাবাদে : 0.7934287633932863
Cosine similarity of  হায়দরাবাদ and  হায়দরাবাদের : 0.7816336196685671
Cosine similarity of  সিবিআই and  সিবিআইকে : 0.7651194952896309
Cosine similarity of  সিবিআই and  সিবিআইয়ের : 0.7777578630294532
Cosine similarity of  করানোর and  করানো : 0.7180158086689256

Words Chosen :
['পাসপোর্টের', 'ইপাসপোর্ট', 'অভিযোগও', 'জালিয়াতি', 'জালিয়াতিসহ', 'হায়দরাবাদের', 'হায়দরাবাদের', 'সিবিআইকে', 'সিবিআইয়ের']
```

Fig. 13 : Chosen similar words using Word Embedding

The effect of word embedding is relative to the query and its corresponding initial search results. For a different query the semantic similar words that are chosen are shown in Fig.14 .



```
Cosine Similarity check between query words and voc words | T>=0.7:
Cosine similarity of  সম্বন্ধে and  সম্পর্কে : 0.7006623005640593
Cosine similarity of  ভারত and  পাকিস্তান : 0.7292054399938865
Cosine similarity of  বিদ্যুৎ and  বিদ্যুত্ : 0.7368209117636956
Cosine similarity of  সরকারের and  সরকার : 0.7806418835906933
Cosine similarity of  সমস্যা and  সমস্যার : 0.7446204285534909
Cosine similarity of  প্রকল্প and  প্রকল্পের : 0.78610896882869
Cosine similarity of  প্রকল্প and  প্রকল্পে : 0.7010647972578059
Cosine similarity of  পাকিস্তান and  ভারত : 0.7292054399938865
Cosine similarity of  পাকিস্তান and  পাকিস্তানে : 0.7377565336782891
Cosine similarity of  পাকিস্তান and  পাকিস্তানের : 0.7775118769452
Cosine similarity of  বিষয় and  বিষয়ই : 0.7257038937568357

Chosen words :

['বিদ্যুত্', 'সরকার', 'সমস্যার', 'প্রকল্পের', 'প্রকল্পে', 'পাকিস্তানে', 'পাকিস্তানের', 'বিষয়ই']
```

Fig. 14 : Another example of using Word Embedding for different query

### 4.4.2.7 Adding Most Frequent Terms from Top Ranked Document(s)

The standard term selection algorithm suggests choosing the most frequent K1 terms from a top ranked document of the Initial Search Results . As an improvement on that we choose the most frequent K1 terms from the top N2 ranked documents of the Initial Search Results.

We set the values of K1 and N2 as follows :

**K1 = 30** and **N2 = 5**

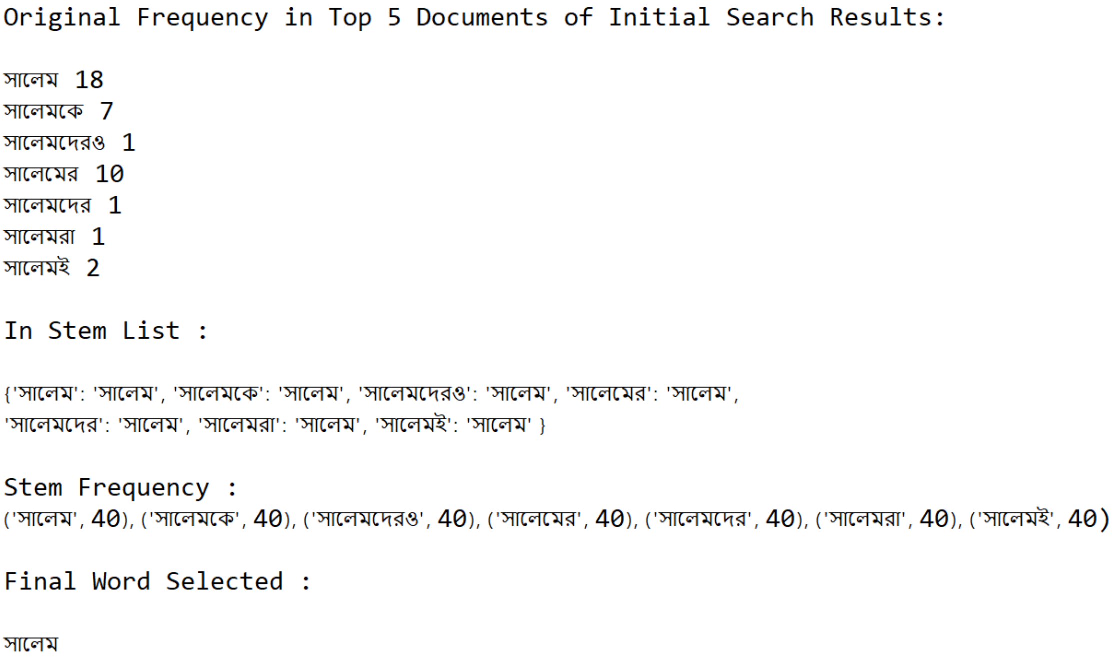
While selecting the top K1 terms from the top N2 ranked documents , the stem frequency of each term is used and all the selected terms have distinct root word , i.e. no two terms that are finally selected have the same stemmed root word .

Stemming is the process of removing affixes from related words in order to reduce them to their stem, base, or root form. Its goal is to convert several derivational or inflectional versions of the same word to a single indexing form.

A stemming method, which reduces all words with the same stem to a common form, is important in many fields of computational linguistics and information retrieval work. For example, the terms "stemmer," "stemming," and "stemmed" may all be reduced to "stem."

Stem Frequency is important since it lowers repetition because most word stems and their inflected/derived words signify the same thing.

An example of Stem Frequency is shown in Fig. 15 :



```
Original Frequency in Top 5 Documents of Initial Search Results:

সালেম  18
সালেমকে  7
সালেমদেরও  1
সালেমের  10
সালেমদের  1
সালেমরা  1
সালেমই  2

In Stem List :

{'সালেম': 'সালেম', 'সালেমকে': 'সালেম', 'সালেমদেরও': 'সালেম', 'সালেমের': 'সালেম',
'সালেমদের': 'সালেম', 'সালেমরা': 'সালেম', 'সালেমই': 'সালেম' }

Stem Frequency :
('সালেম', 40), ('সালেমকে', 40), ('সালেমদেরও', 40), ('সালেমের', 40), ('সালেমদের', 40), ('সালেমরা', 40), ('সালেমই', 40)

Final Word Selected :

সালেম
```

Fig. 15 : An example of Stem Frequency

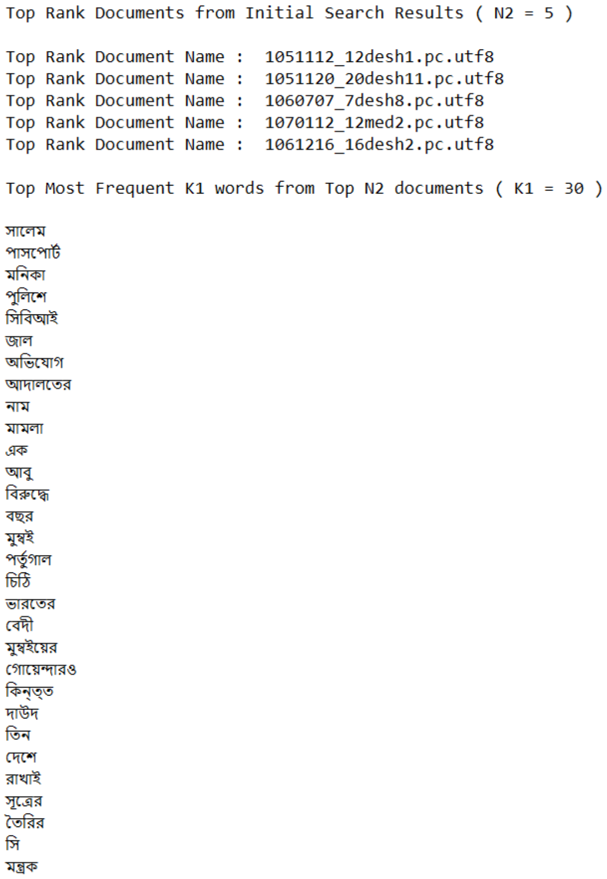
Here , the original frequencies of the Bengali words 'সালেম' , 'সালেমকে' , 'সালেমদেরও' 'সালেমের', ' সালেমদের' , ' সালেমরা' , 'সালেমই' are all different but the root stem word of all the words is 'সালেম', hence the stem frequency of all the words is given as the cumulative of the frequencies of the individual words having the same stem.

Therefore, the stem frequency of the Bengali words 'সালেম' , 'সালেমকে' , 'সালেমদেরও' 'সালেমের', ' সালেমদের' , ' সালেমরা' , 'সালেমই' are given as 40 .

Now since all the words have the same root stem word , only one word is picked .

We have used the stem list provided by FIRE for Bengali IR.

Fig. 16 shows the most frequent words where , K1=30 selected from top ranked documents of Initial Search Results for the said query where , N2=5 , i.e. the top 30 most frequent words are chosen from the top 5 ranked documents of the Initial Search Results. The cumulative stem frequency of distinct terms in those top 5 ranked documents is used while selecting those 30 words.



```
Top Rank Documents from Initial Search Results ( N2 = 5 )

Top Rank Document Name :  1051112_12desh1.pc.utf8
Top Rank Document Name :  1051120_20desh11.pc.utf8
Top Rank Document Name :  1060707_7desh8.pc.utf8
Top Rank Document Name :  1070112_12med2.pc.utf8
Top Rank Document Name :  1061216_16desh2.pc.utf8

Top Most Frequent K1 words from Top N2 documents ( K1 = 30 )

সালেম
পাসপোর্ট
মনিকা
পুলিশে
সিবিআই
জাল
অভিযোগ
আদালতের
নাম
মামলা
এক
আবু
বিরুদ্ধে
বছর
মুম্বই
পর্তুগাল
চিঠি
ভারতের
বেদী
মুম্বইয়ের
গোয়েন্দারও
কিন্‌ত্‌ত
দাউদ
তিন
দেশে
রাখাই
সূত্রের
তৈরির
সি
মন্ত্রক
```

Fig. 16 : Top Ranked Documents (N2=5) and its Most Frequent Words ( K1=30)

These top 30 terms are added to the Candidate Terms Pool .

Say F= the chosen most frequent terms ( Top K1 words from top N2 ranked documents )

$$\text{Candidate Terms Pool} = \text{Candidate Terms Pool} \cup F$$

### 4.4.2.8 Ranking Candidate Terms

Each term t in Candidate Terms Pool is checked for repeats and also stemmed to its root stem word , i.e. no two words now have the same root stem .

The root stem word of each term t in Candidate Terms Pool is checked against each stem word of the pre-processed query terms . If the root stem word for a term t exists in the stemmed list of the pre-processed query terms , then the term t is discarded.

Now , each term in the Candidate Terms Pool is assigned a weight based on the equation 6.

The context score is calculated by finding out the cosine similarity between the word vector of a term t and the mean word vector for the original query terms. For appropriate word vectors , the python package known as FastText is used.

For calculating the mean word vector for original query terms , the average of word vectors of all the query terms after stop word removal is used.

If a word vector for a particular term is not found in the said package , then a zero vector is assigned to it and used.

While calculating the frequency score of a term t , the stem frequency of the term in the document collection (N1=10) is used. It is divided by the maximum stem frequency of a term in that same said document collection.

The value of α ranges from 0 to 1 .

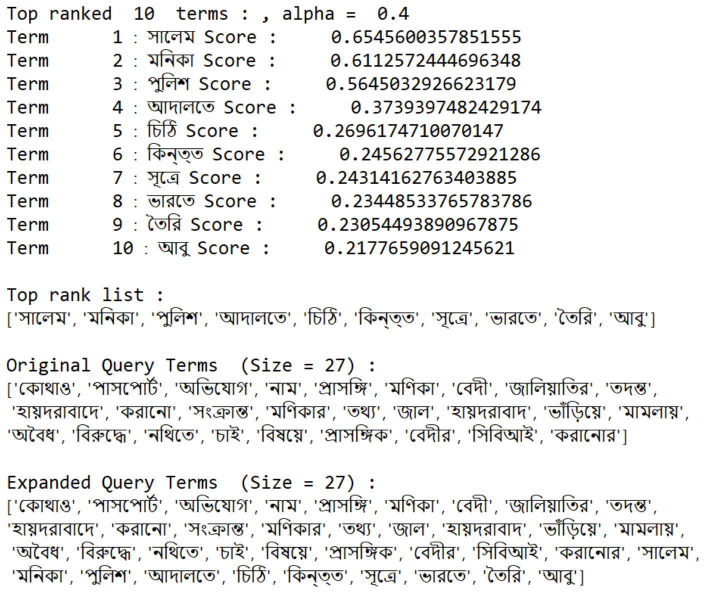
Range of α = [ 0.0 , 0.1 , 0.2 , 0.3 , 0.4 , 0.5 , 0.6 , 0.7 , 0.8 , 0.9 , 1.0 ]

All the terms in a Candidate Terms Pool are assigned weight and sorted according to decreasing weight and from them Top K2 terms are chosen.

It is not possible to find out for which value of α the ranking works out the best before evaluation . Therefore, we rank the query for different values of α and all of them are evaluated . Only after evaluation are we able to find the best value of α .

An example of Candidate Terms Pool ranking is shown in Fig. 17 ,

Where,  K2 = 10 but α = 0.4 .



```
Top ranked  10  terms : , alpha =  0.4
Term      1 : সালেম Score :      0.6545600357851555
Term      2 : মনিকা Score :      0.6112572444696348
Term      3 : পুলিশ Score :      0.5645032926623179
Term      4 : আদালতে Score :      0.3739397482429174
Term      5 : চিঠি Score :      0.2696174710070147
Term      6 : কিন্‌ত্‌ত Score :      0.24562775572921286
Term      7 : সূত্রে Score :      0.24314162763403885
Term      8 : ভারতে Score :      0.23448533765783786
Term      9 : তৈরি Score :      0.23054493890967875
Term      10 : আবু Score :      0.2177659091245621

Top rank list :
['সালেম', 'মনিকা', 'পুলিশ', 'আদালতে', 'চিঠি', 'কিন্‌ত্‌ত', 'সূত্রে', 'ভারতে', 'তৈরি', 'আবু']

Original Query Terms  (Size = 27) :
['কোথাও', 'পাসপোর্ট', 'অভিযোগ', 'নাম', 'প্রাসঙ্গি', 'মণিকা', 'বেদী', 'জালিয়াতির', 'তদন্ত',
 'হায়দরাবাদে', 'করানো', 'সংক্রান্ত', 'মণিকার', 'তথ্য', 'জাল', 'হায়দরাবাদ', 'ভাঁড়িয়ে', 'মামলায়',
'অবৈধ', 'বিরুদ্ধে', 'নথিতে', 'চাই', 'বিষয়ে', 'প্রাসঙ্গিক', 'বেদীর', 'সিবিআই', 'করানোর']

Expanded Query Terms  (Size = 27) :
['কোথাও', 'পাসপোর্ট', 'অভিযোগ', 'নাম', 'প্রাসঙ্গি', 'মণিকা', 'বেদী', 'জালিয়াতির', 'তদন্ত',
'হায়দরাবাদে', 'করানো', 'সংক্রান্ত', 'মণিকার', 'তথ্য', 'জাল', 'হায়দরাবাদ', 'ভাঁড়িয়ে', 'মামলায়',
 'অবৈধ', 'বিরুদ্ধে', 'নথিতে', 'চাই', 'বিষয়ে', 'প্রাসঙ্গিক', 'বেদীর', 'সিবিআই', 'করানোর', 'সালেম',
 'মনিকা', 'পুলিশ', 'আদালতে', 'চিঠি', 'কিন্‌ত্‌ত', 'সূত্রে', 'ভারতে', 'তৈরি', 'আবু']
```

Fig. 17 : First example of Expanded Query

# 4.5 Top K2 terms chosen for Query Expansion

The top K2 terms are chosen by us for query expansion.

It is not possible to find out for which value of K2 the precision for the query increases before evaluation .Therefore , we run evaluation for the query for different values of K2 ranging from 5 to 40 in multiples of 5 , i.e. ,

Range of K2 = [ 5 , 10 , 15 , 20 , 25 , 30 , 35 , 40 ]

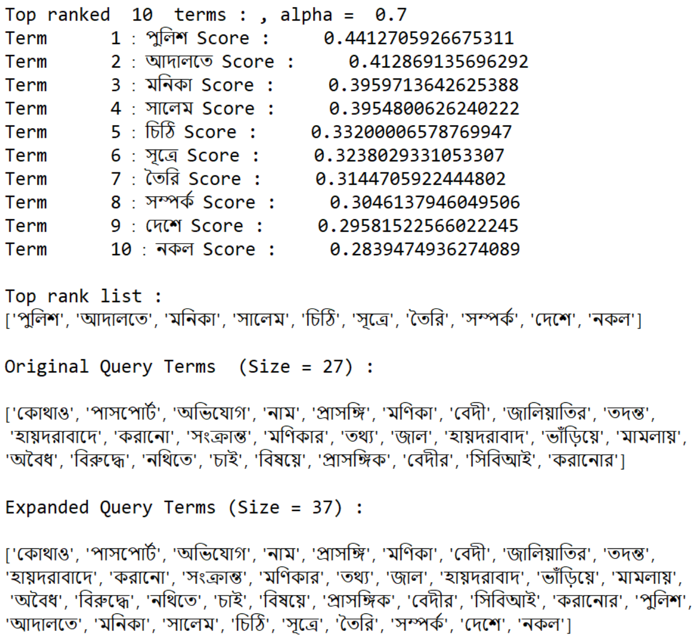
Only after evaluation are we able to find the best value of K2 .

## 4.6 Expanded Query

In this step the original query is reformulated.

The top K2 terms are added to the original query and this list is known as Expanded Query . This procedure is done for all values of α and K2.

Fig. 17 shows the expanded query list after ranking , where K2= 10 and α = 0.4 whereas Fig.18 shows the expanded query list after ranking , where K2= 10 and α = 0.7 . Ranking with different values of alpha does have an effect on the expanded query as shown in Fig. 17 and Fig. 18.



```
Top ranked  10  terms : , alpha =  0.7
Term      1 : পুলিশ Score :     0.4412705926675311
Term      2 : আদালতে Score :     0.412869135696292
Term      3 : মনিকা Score :     0.3959713642625388
Term      4 : সালেম Score :     0.3954800626240222
Term      5 : চিঠি Score :     0.33200006578769947
Term      6 : সূত্রে Score :     0.3238029331053307
Term      7 : তৈরি Score :     0.3144705922444802
Term      8 : সম্পর্ক Score :     0.3046137946049506
Term      9 : দেশে Score :     0.29581522566022245
Term      10 : নকল Score :     0.2839474936274089

Top rank list :
['পুলিশ', 'আদালতে', 'মনিকা', 'সালেম', 'চিঠি', 'সূত্রে', 'তৈরি', 'সম্পর্ক', 'দেশে', 'নকল']

Original Query Terms  (Size = 27) :

['কোথাও', 'পাসপোর্ট', 'অভিযোগ', 'নাম', 'প্রাসঙ্গি', 'মণিকা', 'বেদী', 'জালিয়াতির', 'তদন্ত',
 'হায়দরাবাদে', 'করানো', 'সংক্রান্ত', 'মণিকার', 'তথ্য', 'জাল', 'হায়দরাবাদ', 'ভাঁড়িয়ে', 'মামলায়',
'অবৈধ', 'বিরুদ্ধে', 'নথিতে', 'চাই', 'বিষয়ে', 'প্রাসঙ্গিক', 'বেদীর', 'সিবিআই', 'করানোর']

Expanded Query Terms (Size = 37) :

['কোথাও', 'পাসপোর্ট', 'অভিযোগ', 'নাম', 'প্রাসঙ্গি', 'মণিকা', 'বেদী', 'জালিয়াতির', 'তদন্ত',
'হায়দরাবাদে', 'করানো', 'সংক্রান্ত', 'মণিকার', 'তথ্য', 'জাল', 'হায়দরাবাদ', 'ভাঁড়িয়ে', 'মামলায়',
 'অবৈধ', 'বিরুদ্ধে', 'নথিতে', 'চাই', 'বিষয়ে', 'প্রাসঙ্গিক', 'বেদীর', 'সিবিআই', 'করানোর', 'পুলিশ',
'আদালতে', 'মনিকা', 'সালেম', 'চিঠি', 'সূত্রে', 'তৈরি', 'সম্পর্ক', 'দেশে', 'নকল']
```

Fig. 18 : Second example of Expanded Query

## 4.7 Final Search Results

The reformulated query or the expanded query is rerun it on the same Information Retrieval System to provide us with the Final Search Results.

This procedure is done for all values of K2 and α , so that we could find for which value of K2 and α the precision is best .

The Mean Average Precision or MAP [103] is used to evaluate all the 26 Queries . The MAP of the Initial Search Results is compared with the MAP of the Final Search Results for different combinations of α and K2 .

For a proposed method there are in total 88 combinations of α and K2 , all of them are evaluated using MAP to check for which values of α and K2 the method has the best precision .

The procedure for evaluation , experiments conducted and results obtained for these final search results are discussed in Section 5 .

# 5. EVALUATION,EXPERIMENTS AND RESULTS

## 5.1 Evaluation

To evaluate the IR system , the IR system has been tested against 26 queries to search relevant documents from a corpus of approximately 123021 documents. The IR system performance is measured by the terms of Mean Average Precision (MAP).

The Mean Average Precision (MAP) [102] is used to evaluate the IR models. The MAP evaluation metric requires for each query, the list of ranked documents , i.e. the final search results provided by our IR system , and the list of documents relevant to a query as evaluated by FIRE .

### 5.1.1 Precision

Precision is the fraction of the documents retrieved that are relevant to the user's information need.

Precision , **P** $= \dfrac{Relevant\ Retrieved\ Documents}{Retrieved\ Documents}$ (7)

### 5.1.2 Average Precision

It is desirable to examine the order in which the returned documents are displayed for systems that return a ranked sequence of documents. This measure averages the precision values from the rank positions where a relevant document is retrieved :

**AP** $= \dfrac{\sum_{r=1}^{N} (P(r) * rel\,(r\,))}{C_r}$ (8)

Where :

- **r** is the rank
- **N** is the number of documents retrieved
- **rel(r)** is a binary function on the relevance of a given rank
- **P(r)** is the precision ( proportion of a retrieved set that is relevant) at a given cut-off rank
- $\boldsymbol{C_r}$ is the number of relevant Documents

### 5.1.2 Mean Average Precision ( MAP )

MAP is computed by taking the average of AP over all queries. It summarizes rankings from multiple queries by averaging average precision :

$$\textbf{MAP} \quad = \quad \frac{\sum_{q=1}^{Q} AP\,(\,q\,)}{|Q|} \qquad (9)$$

## 5.2 Experiments

The Mean Average Precision or MAP [102] is used to evaluate all the 26 Queries . The MAP of the Initial Search Results is compared with the MAP of the Final Search Results for different combinations of α and K2 .

For our experiments and evaluation we have proposed seven comparison models . We aim to analyze the MAP scores of these said seven models against the MAP score of the baseline model .

The value of N1 , N2 , K1 is set the same for all models while comparing them. The values are as follows :

**N1 = 10** , **N2 = 5** and **K1 = 30** .

The value of α and K2 are in range :

Range of **α** = [ 0.0 , 0.1 , 0.2 , 0.3 , 0.4 , 0.5 , 0.6 , 0.7 , 0.8 , 0.9 , 1.0 ]

Range of **K2** = [ 5 , 10 , 15 , 20 , 25 , 30 , 35 , 40 ]

For a particular model there are in total 88 combinations of α and K2 , all of them are evaluated using MAP to check for which values of α and K2 the model has the best precision .

Our proposed seven comparison models are as follows :

- **Model Baseline :** Without Query Expansion using Initial Search Results
- **Model A :** Query Expansion adding Synonyms
- **Model B :** Query Expansion adding semantic similar words using Word Embedding from top N2 documents of Initial Search Results
- **Model C :** Query Expansion adding Top K1 terms from top N2 documents of Initial Search Results
- **Model D :** Query Expansion adding synonym + adding Top K1 terms from top N2 documents of Initial Search Results
- **Model E :** Query Expansion adding synonym + adding semantic similar words using Word Embedding from top N2 documents of Initial Search Results
- **Model F :** Query Expansion adding synonym + adding Top K1 terms from top N2 documents of Initial Search Results
- **Model G :** Query Expansion adding synonym + adding semantic similar words using Word Embedding from top N2 documents of Initial Search Results + adding Top K1 terms from top N2 documents of Initial Search Results

The best MAP scores for each model have been calculated and the corresponding values for alpha ( **α** ) and top **K2** terms are noted.

After analyzing the results of the said experiments we are able to find for which model and for which values of **α** and **K2** the proposed Query Expansion method works out the best .

## 5.3 Results

In this section, the results of all the comparison models are presented . The said experiments are conducted and checked for which models the MAP score improves than that of the baseline model .

We found that for five models the MAP scores improve and for two models it does not .

Hence, Query Expansion on Bengali Document indeed increases the MAP for five models. The documents ranked in the final search results are better retrieved than the initial search results.

We are also able to find for which model and their best corresponding values of **α** and **K2** the proposed Query Expansion method works out the best .

The detailed results of our experiments are shown in Table 3.

| **Model Name** | **Alpha ( α )** | **Top K2 Terms** | **MAP** |
|---|---|---|---|
| **Baseline** | | | 0.5136 |
| **Model A** | 0.2 | 5 | 0.5020 |
| **Model B** | 0.3 | 5 | 0.5176 |
| **Model C** | 0.3 | 10 | 0.5363 |
| **Model D** | 0.8 | 5 | 0.5371 |
| **Model E** | 0.7 | 5 | 0.5103 |
| **Model F** | 0.3 | 10 | 0.5404 |
| **Model G** | 0.7 | 5 | **0.5441** |

Table 3 : Results of comparison models

From Table 3 , we infer that the MAP scores for **five** out of our **seven** proposed comparison models are better than the MAP score for our baseline model.

Adding similar words using word embedding and adding top K1 terms from top N2 documents of Initial Search Results , i.e. **Model B** and **Model C** each individually outperforms the scores of the Baseline Model whereas adding synonyms , i.e. **Model A** individually does not .

There are three collaboration models in groups of two , namely Model D and Model E and Model F . The MAP scores for **Model D** and **Model F** is better than that of the Baseline Model whereas The MAP score for **Model E** is not .

Finally, the collaboration of the models A, B, and C , i.e. **Model G** outperforms the baseline model on a higher level, and it is based on the proposed term selection algorithm for Bengali document information retrieval in this thesis in Section 4 .

This means that adding synonyms and similar semantic words using a word embedding together is not enough to outperform the baseline model but adding similar words using word embedding and adding top K1 terms from top N2 documents of Initial Search Results is enough to outperform the baseline model. Hence, we can see that top K1 terms are vital for our proposed method.

These experiments show that using the Pseudo Relevance Feedback approach of Query Expansion is effective. Selecting N1 and N2 top ranked documents of the initial search results to choose semantic similar words from N1 and top most K1 frequent terms from N2 , respectively is beneficial .

After analyzing the comparison models , we see that the proposed hybrid collaboration model, **Model G** gives us the best MAP of **0.5441** for the final search results as compared to the baseline model which gives us the MAP of 0.5136 for the initial search results.

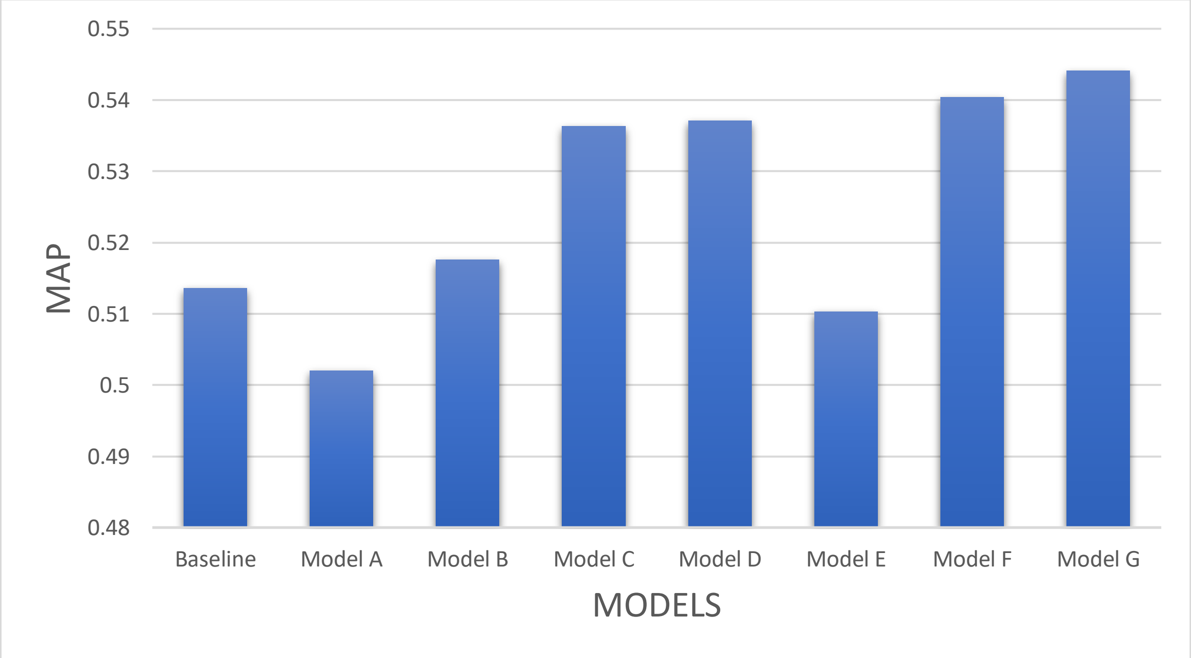
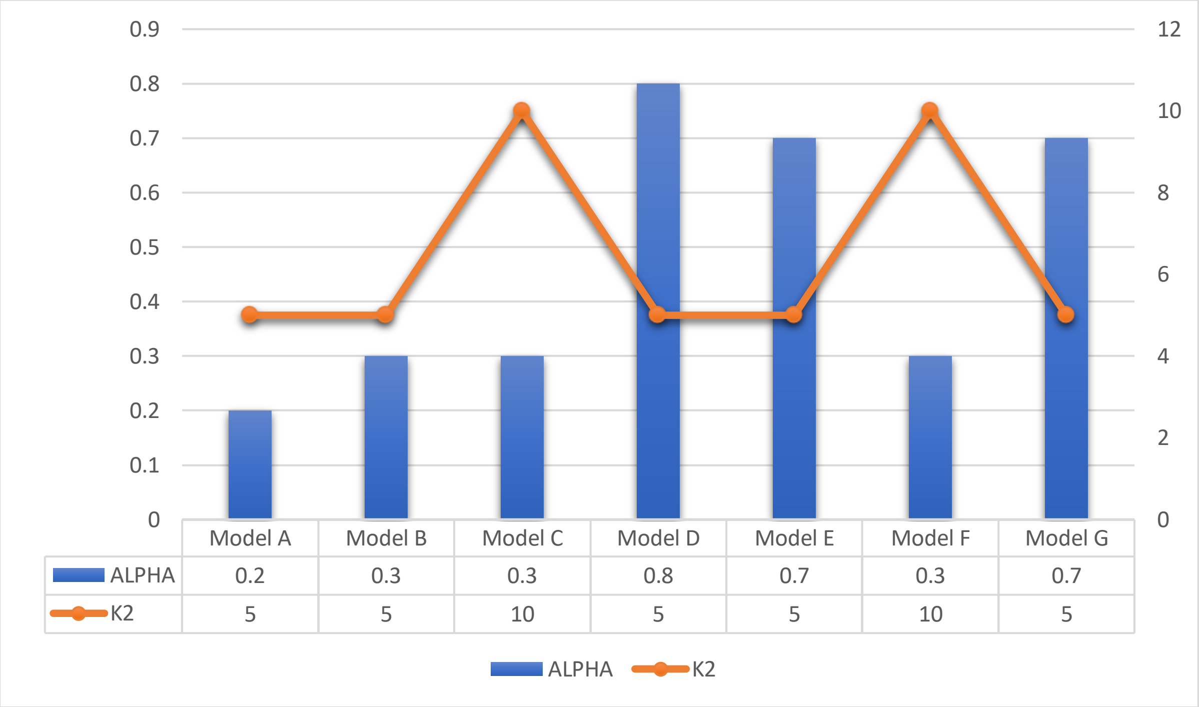


Fig. 19 : MAP for baseline and seven comparison models

Fig. 19 shows the bar chart of MAP scores of the baseline model and the seven comparison models.



| | Model A | Model B | Model C | Model D | Model E | Model F | Model G |
|---|---|---|---|---|---|---|---|
| ALPHA | 0.2 | 0.3 | 0.3 | 0.8 | 0.7 | 0.3 | 0.7 |
| K2 | 5 | 5 | 10 | 5 | 5 | 10 | 5 |

Fig. 20 : Alpha and K2 for seven comparison models

The best values of tuning parameters Alpha (α) and K2 differ for different comparison models. For any particular comparison model the values of Alpha and K2 vary in their proposed ranges and only the values corresponding to the best MAP are noted as shown in Fig. 20 .

As the MAP of Model B , Model C , Model D , Model F and Model G beats the MAP of the baseline model , we analyze these five models for finding out probable ranges of α and K2 **.**

For all the evaluated 26 Queries, we see that **5** to **10** top **K2** terms are enough for effective Query Expansion and increased MAP and the range of **α** for effective document retrieval is **0.3** to **0.8** .

Thus , our proposed hybrid approach for Bengali document Information Retrieval using Query Expansion which implements the Vector Space Paradigm of Information Retrieval and uses the Pseudo Relevance Feedback approach of  Query Expansion indeed helps in better retrieval of Bengali documents.

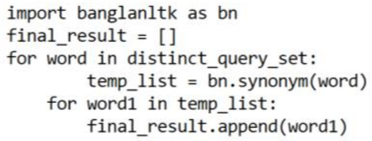
# 6. IMPLEMENTATION

The proposed Bengali document IR using Query Expansion is developed in **Python 3.7** and built with the PyCharm integrated development environment.

The **Bengali IR** system was created in-house with our own code to get the Initial Search Results.

Our term selection method for QE is divided into three stages, with each portion implemented as follows:

1. **Synonyms Set** :
   For our synonyms set we have used a python package named Bangla NLTK ,which has an MIT License.
   Source Link : https://pypi.org/project/banglanltk/
   Code Snippet :



```
import banglanltk as bn
final_result = []
for word in distinct_query_set:
        temp_list = bn.synonym(word)
    for word1 in temp_list:
        final_result.append(word1)
```

Fig. 21 : A code snippet of Synonym package

2. **Semantic Set** :
   For appropriate word vectors we have used a python package known as FastText, which is regulated and maintained by Meta.
   Source Link : https://fasttext.cc/
   FastText contains word vector models for 157 languages[104] , we utilized the one for Bengali .
   Bengali Word Vector Model Source Link :
   https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.bn.300.vec.gz

3. **Most Frequent Set** :
   For our most frequent set we developed our own code as we could not find any helpful resource.

To get the Final Search Results , we performed Query Expansion using our proposed term selection method and rerun it on the said in-house Bengali IR system . This process of QE and also the evaluation was developed by us using our own code.

# 7. CONCLUSION

In this thesis , we have proposed a hybrid solution to Bengali Document Information Retrieval using Query Expansion , which implements the Vector Space Paradigm of Information Retrieval and leverages the Pseudo Relevance Feedback technique of Query Expansion.

Experiments on comparison models demonstrate that the suggested strategy outperforms the unexpanded baseline model.

The MAP scores for five of our seven proposed comparison models are higher than our baseline model's MAP score.

The MAP for the final search results of **Model G**, using QE is **0.5441** whereas the MAP for our baseline model on initial search results without using QE is **0.5136** , thus showing that QE indeed helps better retrieve Bengali documents on FIRE ad hoc dataset 2010.

Hence , Bengali Document Information Retrieval can be improved using the local analysis approach of Query Expansion using Pseudo Relevance Feedback .

The reformulation of the query using our proposed term selection algorithm for Bengali Information Retrieval is thus useful to provide us with better final search results.

The best value of alpha differs for different comparison models and is usually in the range of 0.3 to 0.8 but the best values of K2 are in the range of **5-10** terms as seen in Fig. 17 , i.e. Query Expansion using Pseudo Relevance Feedback for Bengali document IR works best if the 5-10 terms are finally added to the initial query for query reformulation as per our proposed term selection algorithm for Bengali IR .

Therefore the collaboration of Models A,B and C , i.e. Model G , implemented as per our proposed term selection algorithm for Bengali IR gives us the best results. The collaboration model , Model G, adds synonyms , adds semantic similar words using Word Embedding from top N2 documents of Initial Search Results and adds Top K1 terms from top N2 documents of Initial Search Results.

According to our best Model G , the best values of **α** and **K2** are **0.7** and **5** respectively.

The retrieval accuracy of the hybrid model , Model G may be enhanced further by employing a better Bengali stem list, better word vectors for Bengali words, and a better Bengali synonym generator. Instead of adopting the stem list supplied by FIRE, one can create their own stemmer to create a suitable stem list. The word vectors may be enhanced by employing an even larger corpus than the current one to compute word vectors, hence enhancing the word embedding of the proposed Bengali document IR model. Instead of incorporating the existing open-source synonym generator, one can create their own Bengali synonym generator from the scratch.

When models are combined, the hybrid system becomes relatively slow. One can also investigate the issues related to the speed of the hybrid model in the future .

Thus, future works might include investigating the speed of the IR system, developing a better stemmer and a synonym generator for Bengali IR, and improving word vectors for Bengali words , all for the sole purpose of increasing the retrieval accuracy of the proposed hybrid model.

# REFERENCES

1. Jurafsky, D., & Martin, J. (2008). Speech and Language Processing (2nd ed.). Pearson.
2. Chatterjee, S., & Sarkar, K. (2018). Combining IR Models for Bengali Information Retrieval. *International Journal of Information Retrieval Research (IJIRR)*, *8*(3), 68-83.
3. Manning, C. D., Raghavan, P., & Schutze, H. (2012). Introduction to Information Retrieval. Cambridge University Press. https://doi.org/10.1017/cbo9780511809071
4. Query expansion - Wikipedia. (n.d.). Query Expansion - Wikipedia; en.wikipedia.org. Retrieved June 7, 2022, from https://en.wikipedia.org/wiki/Query_expansion
5. Azad, H. K., & Deepak, A. (2019). Query expansion techniques for information retrieval: a survey. *Information Processing & Management*, *56*(5), 1698-1735.
6. Luhn, H. P. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of research and development*, *1*(4), 309-317.
7. Salton, G., Singhal, A., Mitra, M., & Buckley, C. (1997). Automatic text structuring and summarization. *Information processing & management*, *33*(2), 193-207.
8. Xu, J., & Croft, W. B. (2017, August). Query expansion using local and global document analysis. In Acm sigir forum (Vol. 51, No. 2, pp. 168-175). New York, NY, USA: ACM.
9. Bouadjenek, M. R., Hacid, H., Bouzeghoub, M., & Daigremont, J. (2011, July). Personalized social query expansion using social bookmarking systems. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval* (pp. 1113-1114)
10. Biancalana, C., & Micarelli, A. (2009, August). Social tagging in query expansion: A new way for personalized web search. In *2009 International Conference on Computational Science and Engineering* (Vol. 4, pp. 1060-1065). IEEE.
11. Zhou, D., Lawless, S., & Wade, V. (2012). Improving search via personalized query expansion using social media. *Information retrieval*, *15*(3), 218-242.
12. Lin, D., & Pantel, P. (2001). Discovery of inference rules for question-answering. *Natural Language Engineering*, *7*(4), 343-360.
13. Agichtein, E., Lawrence, S., & Gravano, L. (2004). Learning to find answers to questions on the web. *ACM Transactions on Internet Technology (TOIT)*, *4*(2), 129-162.
14. Riezler, S., Liu, Y., & Vasserman, A. (2008, August). Translating queries into snippets for improved query expansion. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)* (pp. 737-744).
15. Wu, H., Wu, W., Zhou, M., Chen, E., Duan, L., & Shum, H. Y. (2014, February). Improving search relevance for short queries in community question answering. In *Proceedings of the 7th ACM international conference on Web search and data mining* (pp. 43-52).
16. Shekarpour, S., Höffner, K., Lehmann, J., & Auer, S. (2013, September). Keyword query expansion on linked data using linguistic and semantic features. In *2013 IEEE seventh international conference on semantic computing* (pp. 191-197). IEEE.

17. Pirkola, A., Hedlund, T., Keskustalo, H., & Järvelin, K. (2001). Dictionary-based cross-language information retrieval: Problems, methods, and research findings. *Information retrieval*, *4*(3), 209-230.
18. Ballesteros, L., & Croft, W. B. (1997, July). Phrasal translation and query expansion techniques for cross-language information retrieval. In *ACM SIGIR Forum* (Vol. 31, No. SI, pp. 84-91). New York, NY, USA: ACM.
19. Kraaij, W., Nie, J. Y., & Simard, M. (2003). Embedding web-based statistical translation models in cross-language information retrieval. *Computational Linguistics*, *29*(3), 381-419.
20. Gaillard, B., Bouraoui, J. L., De Neef, E. G., & Boualem, M. (2010, May). Query expansion for cross language information retrieval improvement. In *2010 Fourth International Conference on Research Challenges in Information Science (RCIS)* (pp. 337-342). IEEE.
21. Ballesteros, L., & Croft, W. B. (1998, August). Resolving ambiguity for cross-language retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 64-71).
22. Ballesteros, L. A. (2002). Cross-language retrieval via transitive translation. In *Advances in information retrieval* (pp. 203-234). Springer, Boston, MA.
23. Levow, G. A., Oard, D. W., & Resnik, P. (2005). Dictionary-based techniques for cross-language information retrieval. *Information processing & management*, *41*(3), 523-547.
24. McNamee, P., & Mayfield, J. (2002, August). Comparing cross-language query expansion techniques by degrading translation resources. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 159-166).
25. Allan, J. (1996, August). Incremental relevance feedback for information filtering. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 270-278).
26. Eichstaedt, M., Patel, A. P., Lu, Q., Manber, U., & Rudkin, K. (2002). *U.S. Patent No. 6,381,594*. Washington, DC: U.S. Patent and Trademark Office.
27. Wu, Y., Liu, X., Xie, M., Ester, M., & Yang, Q. (2016, February). CCCF: Improving collaborative filtering via scalable user-item co-clustering. In *Proceedings of the ninth ACM international conference on web search and data mining* (pp. 73-82).
28. Singhal, A., & Pereira, F. (1999, August). Document expansion for speech retrieval. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 34-41).
29. Johnson, S. E., Jourlin, P., Jones, K. S., & Woodland, P. C. (2000, March). Spoken Document Retrieval for TREC-9 at Cambridge University. In *TREC*.
30. Natsev, A., Haubold, A., Tešić, J., Xie, L., & Yan, R. (2007, September). Semantic concept-based query expansion and re-ranking for multimedia retrieval. In *Proceedings of the 15th ACM international conference on Multimedia* (pp. 991-1000).
31. Kuo, Y. H., Chen, K. T., Chiang, C. H., & Hsu, W. H. (2009, October). Query expansion for hash-based image object retrieval. In *Proceedings of the 17th ACM international conference on Multimedia* (pp. 65-74).

32. Hua, X. S., Yang, L., Wang, J., Wang, J., Ye, M., Wang, K., ... & Li, J. (2013, October). Clickage: Towards bridging semantic and intent gaps via mining click logs of search engines. In *Proceedings of the 21st ACM international conference on Multimedia* (pp. 243-252).
33. Xie, H., Zhang, Y., Tan, J., Guo, L., & Li, J. (2014). Contextual query expansion for image retrieval. *IEEE Transactions on Multimedia*, *16*(4), 1104-1114.
34. Islam, M. R., Rahman, J., Talha, M. R., & Chowdhury, F. (2020, June). Query Expansion for Bangla Search Engine Pipilika. In *2020 IEEE Region 10 Symposium (TENSYMP)* (pp. 1367-1370). IEEE.
35. Rasheed, I., & Banka, H. (2018, March). Query expansion in information retrieval for Urdu language. In *2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP)* (pp. 1-6). IEEE.
36. Babu, A., & Sindhu, L. (2015, August). An information retrieval system for Malayalam using query expansion technique. In *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1559-1564). IEEE.
37. Nawab, R. M. A., Stevenson, M., & Clough, P. (2016). An IR-based approach utilizing query expansion for plagiarism detection in MEDLINE. *IEEE/ACM transactions on computational biology and bioinformatics*, *14*(4), 796-804.
38. Atefeh, F., & Khreich, W. (2015). A survey of techniques for event detection in twitter. *Computational Intelligence*, *31*(1), 132-164.
39. de Boer, M., Schutte, K., & Kraaij, W. (2016). Knowledge based query expansion in complex multimedia event detection. *Multimedia Tools and Applications*, *75*(15), 9025-9043.
40. Zhang, Z., Wang, Q., Si, L., & Gao, J. (2016, July). Learning for efficient supervised query expansion via two-stage feature selection. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval* (pp. 265-274).
41. Magdy, W., & Jones, G. J. (2011, October). A study on query expansion methods for patent retrieval. In *Proceedings of the 4th workshop on Patent information retrieval* (pp. 19-24).
42. Huber, S., Seiger, R., Kühnert, A., Theodorou, V., & Schlegel, T. (2016). Goal-based semantic queries for dynamic processes in the internet of things. *International Journal of Semantic Computing*, *10*(02), 269-293.
43. Abdulla, A. A. A., Lin, H., Xu, B., & Banbhrani, S. K. (2016). Improving biomedical information retrieval by linear combinations of different query expansion techniques. *BMC bioinformatics*, *17*(7), 443-454.
44. Liu, X., Chen, F., Fang, H., & Wang, M. (2014). Exploiting entity relationship for query expansion in enterprise search. *Information retrieval*, *17*(3), 265-294.
45. Nie, L., Jiang, H., Ren, Z., Sun, Z., & Li, X. (2016). Query expansion based on crowd knowledge for code search. *IEEE Transactions on Services Computing*, *9*(5), 771-783.
46. MacFarlane, A., Robertson, S. E., & McCann, J. A. (1997). Parallel computing in information retrieval–an updated review. *Journal of Documentation*.
47. Zingla, M. A., Chiraz, L., & Slimani, Y. (2016). Short query expansion for microblog retrieval. *Procedia Computer Science*, *96*, 225-234.

48. Marcus, R. S. (1991). Computer and Human Understanding in Intelligent Retrieval Assistance. In *Proceedings of the ASIS Annual Meeting* (Vol. 28, pp. 49-59).
49. Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613-620.
50. Ponte, J. M., & Croft, W. B. (2017, August). A language modeling approach to information retrieval. In *ACM SIGIR Forum* (Vol. 51, No. 2, pp. 202-208). New York, NY, USA: ACM.
51. Rocchio, J. J. (1965). Relevance Feedback in Information Retrieval, Report No. *ISR-9 to the National Science Foundation, The Computation Laboratory of Harvard University, to appear August*.
52. Jones, K. S. (1971). Automatic keyword classification for information retrieval.
53. Van Rijsbergen, C. J. (1977). A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of documentation*.
54. Krovetz, R., & Croft, W. B. (1992). Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems (TOIS)*, *10*(2), 115-141.
55. Efthimiadis, E. N. (1996). Query Expansion. *Annual review of information science and technology (ARIST)*, *31*, 121-87.
56. Robertson, A. M., & Willett, P. (1993). A comparison of spelling-correction methods for the identification of word forms in historical text databases. *Literary and linguistic computing*, *8*(3), 143-152.
57. Amati, G., & Joost, C. (2003). Van Rijsbergen. *Probabilistic models for information retrieval based on divergence from randomness*, *26*, 27.
58. Chang, Y., Ounis, I., & Kim, M. (2006). Query reformulation using automatically generated query concepts from a document space. *Information processing & management*, *42*(2), 453-468.
59. Harman, D. (1992, June). Relevance feedback revisited. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 1-10).
60. Paik, J. H., Pal, D., & Parui, S. K. (2014). Incremental blind feedback: An effective approach to automatic query expansion. *ACM Transactions on Asian Language Information Processing (TALIP)*, *13*(3), 1-22.
61. Bernardini, A., & Carpineto, C. (2008). *Fub at trec 2008 relevance feedback track: extending rocchio with distributional term analysis*. FONDAZIONE UGO BORDONI ROME (ITALY).
62. Buckley, C., Salton, G., Allan, J., & Singhal, A. (1995). Automatic query expansion using SMART: TREC 3. *NIST special publication sp*, 69-69.
63. Sihvonen, A., & Vakkari, P. (2004, April). Subject Knowledge, Thesaurus-assisted Query Expansion and Search Success. In *RIAO* (Vol. 2004, pp. 393-404).
64. Carpineto, C., De Mori, R., Romano, G., & Bigi, B. (2001). An information-theoretic approach to automatic query expansion. *ACM Transactions on Information Systems (TOIS)*, *19*(1), 1-27
65. Billerbeck, B., & Zobel, J. (2004, January). Questioning query expansion: An examination of behaviour and parameters. In *Proceedings of the 15th Australasian database conference-Volume 27* (pp. 69-76).
66. Salton, G., & Buckley, C. (1990). Improving retrieval performance by relevance feedback. *Journal of the American society for information science*, *41*(4), 288-297.

67. Rocchio, J. (1971). Relevance feedback in information retrieval. *The Smart retrieval system-experiments in automatic document processing*, 313-323.
68. Porter, M. F. (1980). An algorithm for suffix stripping. *Program*.
69. Hull, D. A. (1996). Stemming algorithms: A case study for detailed evaluation. *Journal of the American Society for Information Science*, *47*(1), 70-84.
70. Bhogal, J., MacFarlane, A., & Smith, P. (2007). A review of ontology based query expansion. *Information processing & management*, *43*(4), 866-886.
71. Wu, J., Ilyas, I., & Weddell, G. (2011). A study of ontology-based query expansion. In *Technical report CS-2011–04*.
72. Zhang, Y., & Clark, S. (2011). Syntactic processing using the generalized perceptron and beam search. *Computational linguistics*, *37*(1), 105-151.
73. Crouch, C. J., & Yang, B. (1992, June). Experiments in automatic statistical thesaurus construction. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 77-88).
74. Natsev, A., Haubold, A., Tešić, J., Xie, L., & Yan, R. (2007, September). Semantic concept-based query expansion and re-ranking for multimedia retrieval. In *Proceedings of the 15th ACM international conference on Multimedia* (pp. 991-1000).
75. Kuzi, S., Shtok, A., & Kurland, O. (2016, October). Query expansion using word embeddings. In *Proceedings of the 25th ACM international on conference on information and knowledge management* (pp. 1929-1932).
76. Cui, H., Wen, J. R., Nie, J. Y., & Ma, W. Y. (2002, May). Probabilistic query expansion using query logs. In *Proceedings of the 11th international conference on World Wide Web* (pp. 325-332).
77. Huang, C. K., Chien, L. F., & Oyang, Y. J. (2003). Relevant term suggestion in interactive web search based on contextual information in query session logs. *Journal of the American Society for Information Science and Technology*, *54*(7), 638-649.
78. Baeza-Yates, R., & Tiberi, A. (2007, August). Extracting semantic relations from query logs. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 76-85).
79. McBryan, O. A. (1994, May). GENVL and WWWW: Tools for taming the web. In *Proceedings of the first international world wide web conference* (Vol. 341).
80. Arguello, J., Elsas, J. L., Callan, J., & Carbonell, J. (2008). Document representation and query expansion models for blog recommendation. In *Proceedings of the International AAAI Conference on Web and Social Media* (Vol. 2, No. 1, pp. 10-18).
81. Karan, M., & Šnajder, J. (2015, September). Evaluation of manual query expansion rules on a domain specific faq collection. In *International Conference of the Cross-Language Evaluation Forum for European Languages* (pp. 248-253). Springer, Cham.
82. Chirita, P. A., Firan, C. S., & Nejdl, W. (2007, July). Personalized query expansion for the web. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 7-14).
83. Gao, G., Liu, Y. S., Wang, M., Gu, M., & Yong, J. H. (2015). A query expansion method for retrieving online BIM resources based on Industry Foundation Classes. *Automation in construction*, *56*, 14-25.

84. Croft, W. B., & Harper, D. J. (1979). Using probabilistic models of document retrieval without relevance information. *Journal of documentation*.
85. Xu, J., & Croft, W. B. (2000). Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems (TOIS)*, *18*(1), 79-112.
86. Xu, J., & Croft, W. B. (1996). Query expansion using local and global document analysis In: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval.
87. Lam-Adesina, A. M., & Jones, G. J. (2001, September). Applying summarization techniques for term selection in relevance feedback. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 1-9).
88. Chang, Y., Ounis, I., & Kim, M. (2006). Query reformulation using automatically generated query concepts from a document space. *Information processing & management*, *42*(2), 453-468.
89. Sarkar, K., & Gupta, A. (2017). An empirical study of some selected ir models for Bengali monolingual information retrieval. *arXiv preprint arXiv:1706.03266*.
90. Dolamic, L., & Savoy, J. (2009). Indexing and stemming approaches for the Czech language. Information Processing & Management, 45(6), 714-720.
91. Paik, J. H., & Parui, S. K. (2008). A simple stemmer for inflectional languages. In *Forum for Information Retrieval Evaluation*.
92. Pakray, P., Bhaskar, P., Banerjee, S., Pal, B. C., Bandyopadhyay, S., & Gelbukh, A. F. (2011, September). A Hybrid Question Answering System based on Information Retrieval and Answer Validation. In *CLEF (Notebook Papers/Labs/Workshop)* (Vol. 96).
93. Bhaskar, P., Das, A., Pakray, P., & Bandyopadhyay, S. (2010). Theme based english and bengali ad-hoc monolingual information retrieval in fire 2010. *Corpus*, *1*, 25-586.
94. Ganguly, D., Leveling, J., & Jones, G. J. (2013). Overview of the personalized and collaborative information retrieval (PIR) track at FIRE-2011. In Multilingual Information Access in South Asian Languages (pp. 227-240). Springer, Berlin, Heidelberg.
95. Loponen, A., Paik, J., & Jarvelin, K. (2010). 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)*.
96. Ganguly, D., Leveling, J., & Jones, G. J. (2013, September). A case study in decompounding for Bengali information retrieval. In *International Conference of the Cross-Language Evaluation Forum for European Languages* (pp. 108-119). Springer, Berlin, Heidelberg.
97. Barman, A. K., Sarmah, J., & Sarma, S. K. (2013, April). WordNet based information retrieval system for assamese. In *2013 UKSim 15th International Conference on Computer Modelling and Simulation* (pp. 480-484). IEEE.
98. Chandra, G., & Dwivedi, S. K. (2020). Query expansion based on term selection for Hindi–English cross lingual IR. *Journal of King Saud University-Computer and Information Sciences*, *32*(3), 310-319.

99. Roy, D., Paul, D., Mitra, M., & Garain, U. (2016). Using word embeddings for automatic query expansion. *arXiv preprint arXiv:1606.07608*.

100. Robertson, S. E., Walker, S., Jones, S., & Hancock-Beaulieu, M. M. Gatford (1995). Okapi at trec-3. In *Proceedings of the Third Text REtrieval Conference (TREC-3)*.

101. Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval*, *3*(4), 333-389.

102. Schütze, H., Manning, C. D., & Raghavan, P. (2008). *Introduction to information retrieval* (Vol. 39, pp. 234-265). Cambridge: Cambridge University Press.

103. Schütze, H., Manning, C. D., & Raghavan, P. (2008). *Introduction to information retrieval* (Vol. 39, pp. 234-265). Cambridge: Cambridge University Press.

104. Grave, E., Bojanowski, P., Gupta, P., Joulin, A., & Mikolov, T. (2018). Learning word vectors for 157 languages. arXiv preprint arXiv:1802.06893.

.