BMS COLLEGE OF ENGINEERING

(Autonomous College under VTU)

Bull Temple Road, Basavanagudi, Bangalore - 560019

A project report on

"Information Retrieval"

Submitted in partial fulfillment of the requirements for the award of degree

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE AND ENGINEERING

By

Prateek Jain	**Hema M S**	**Vineeth kumar G**
1BM18IS149	**1BM18IS043**	**1BM18IS036**

Under the guidance of

Anitha H M

Assistant Professor,ISE,BMSCE

Department of Information Science and Engineering

2020-21

BMS COLLEGE OF ENGINEERING
(Autonomous College under VTU)
Bull Temple Road, Basavanagudi,
Bangalore – 560019

Department of Information Science and Engineering

C E R T I F I C A T E

This is to certify that the project entitled "***Information Retrieval***" is a bona-fide work carried out by **Prateek Jain(1BM18IS149), Hema M S(1BM18IS043),Vineeth Kumar G(1BM18IS036)** , in partial fulfillment for the award of degree of Bachelor of Engineering in **Information Science and Engineering** from **Visvesvaraya Technological University, Belgaum** during the year **2020-21**. It is certified that all corrections/suggestions indicated for Internal Assessments have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the Bachelor of Engineering Degree.

Guide Name: Anitha H M
Designation: Assistant Professor

Signatures

Guide	**Professor and HOD**	**Principal**

Examiners

Name of the Examiner	**Signature of the Examiner**

Table of Contents

SL.NO	TOPICS	PAGE NO.
1	Abstract	4
2	Introduction	5
3	Problem Statement	6
4	Tools and Data Structure	7
5	Literature survey	8
6	Base Work	22
7	Comparative Analysis	38
8	High Level Design	40
9	Functional and Non functional Req	42
10	Implementation	43
11	Testing	49
12	Results and analysis	50
13	Conclusion	54
14	Future Work	55
15	References	56

Abstract

Information Retrieval (IR) is the domain that deals with retrieval of unstructured data, especially textual documents, in response to a query. The need for effective methods of automated IR has grown in importance because of the tremendous amount of data and retrieval of relevant information quickly and effectively.

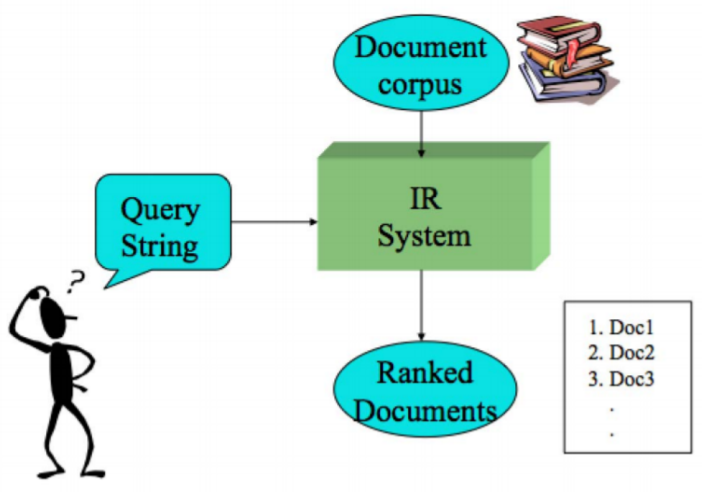
An Information Retrieval System is capable of storage, retrieval, and maintenance of an Information.Information can be text, images, audio, video and other multimedia objects. The TF-IDF weight is a statistical measure that is used to evaluate how important a word is to a document corpus.

There are various models for weighting terms of corpus documents and query terms.The work is carried out to analyze and evaluate the retrieval effectiveness of vector space model(VSM) using email dataset as corpus from kaggle.

Introduction

Many applications that handle information on the internet would be completely inadequate without the support of information retrieval technology. How would we find information on the world wide web if there were no web search engines?

How would we manage our emails without spam filtering? Much of the development of information retrieval technology, such as web search engines and spam filters, requires a combination of experimentation and theory.



Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

An information retrieval (IR) system is a set of algorithms that facilitate the relevance of displayed documents to searched queries. In simple words, it works to sort and rank documents based on the queries of a user.

Problem Statement

- Analysing various methods in the IR model and selecting the best suitable for the domain.
- Finding the most required information using Information retrieval methods from the dataset.

Objective

- The task is to build a search engine which will cater to the needs of a particular domain.We have to feed the IR model with documents containing information about the chosen domain. It will then process the data and build indexes. Once this is done, the user will give a query as an input. It is supposed to return top 10 relevant documents as the output.

Tools and Data Structure

Tools

- ❑ Python platform:Microsoft visual studios/Sublime text/Anaconda/Jupyter Notebook/Google collab.

Data Structures used:

- ❑ Lists in Python
- ❑ Dictionaries in python

Dataset:

- ❑ The corpus is the set of emails received by different people

Important Library:

- ❑ Nltk

Literature Survey :

[1] Information Retrieval Techniques and Applications

- Information retrieval is generally considered as a subfield of computer science that deals with the representation, storage, and access of information.
- **Indexing Techniques :**

  Signature File - The signatures are stored in a separate file called signature file so that search becomes faster and also easy.

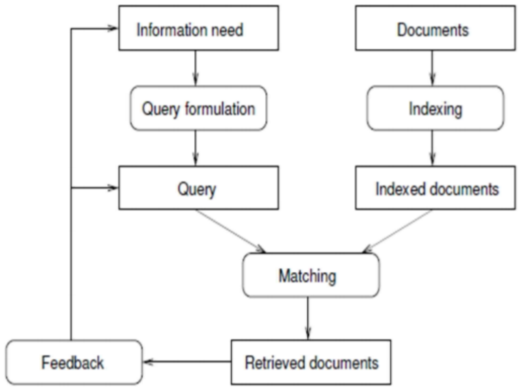
  Inversion Indices - It stores mapping from word to document.

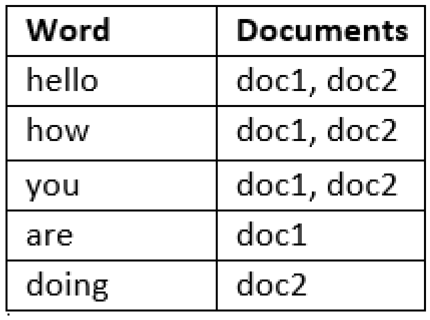
  Ex : doc1 : "hello, how are you"; doc2 : "hello, how you doing"

- **Applications :**

  Digital Library - The information will be stored in digital format and they can be accessed using computers.

  Search Engine and Media Search - This includes searching of media,audio and video files etc





Word	Documents
hello	doc1, doc2
how	doc1, doc2
you	doc1, doc2
are	doc1
doing	doc2

Fig : Information retrieval model Fig : Inversion Indices

[2] A Survey in Traditional Information Retrieval Models

- **Set Theoretic Models :**

  Boolean Model - Boolean algorithm is based on set theory and Boolean algebra. Ex : searching like cats **AND** dogs instead of searching only cats or only dogs.Using this the time used for searching is reduced.

  Case-based reasoning (CBR) Model - CBR model is based on four main methods : Retrieve,Reuse,Revise,Retain.

- **Algebraic Models :**

  Vector Space Model(VSM) - In VSM, the process of indexing is done using TF-IDF model, each document is represented as vector which includes document in which the word is present, term frequency of that word and also inverse document frequency of that

  Neural Network Model - This model includes three layers : The query terms, The document terms and The documents**.**

- **Probabilistic Model :** This model gives the probability with which the given input can be found in the given documents.

[3]Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents

TF-IDF is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.

Procedure and Methods

- **Data Collection -** The data was collected from 20 different random websites that belongs to 4 different domains.
- **Pre-processing -** Preprocessing involves processes like tokenization, stemming, lemmatization, stop word removal and punctuation removal.
- **Design -** Design involves following steps
    - ➔ Count the total number of words in every document and calculate the term frequency of each word in every document.
    - ➔ Counting the total number of documents and checking if the word is present in how many documents and later calculating the inverse document frequency.
    - ➔ Finally multiplying term frequency and inverse document frequency to find TF-IDF

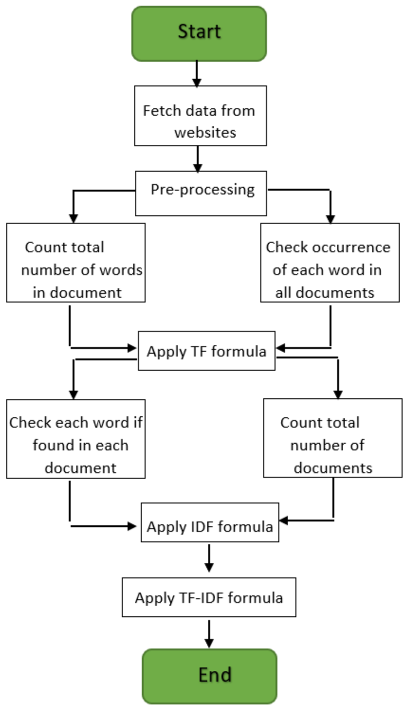

Start
Fetch data from websites
Pre-processing
Count total number of words in document
Check occurrence of each word in all documents
Apply TF formula
Check each word if found in each document
Count total number of documents
Apply IDF formula
Apply TF-IDF formula
End

Fig : Proposed method

[4] Hash Join Algorithms Used in Text-Based Information Retrieval

To implement queries is Hashing different types of hashing algorithms are used in IR. Algorithms discussed in this paper are varied based on its techniques involved in join operations.

1. **Join Operation**- Facilitates the retrieval of the information from two different relations based on aCartesian product of the two relations based on their related tuples.

   1.1 Nested-Loops Join Method - One of the relations is counted as the inner relation, while the other one is counted as the outer one.

   1.2 Sort-Merge Join Method - It's a blocking algorithm. It requires the input relation to be almost fully sorted before the actual join processing starts.

   1.3 Hash-Based Join Method - Reduces the number of comparisons between tuples as the sort-merge join algorithms.

2. **XJoin Algorithm** - XJoin extends Symmetric Hash Join to use less memory by allowing parts of the hash tables to be moved to a secondary storage through partitioning its inputs.

   When a tuple arrives from one source, it is used to probe the other source in event there is enough memory available

3. **Early Hash Join (EHJ) algorithm** - Buffer is used which stores the retrieved data of one relation in memory as possible.

   EHJ improves on the Hybrid Hash Join algorithm in terms of the **flushing policy**.

4. **Hash Merge Join (MHJ) algorithm** - HMJ algorithm involves two phases , which are hashing and merging. The flushing policy used in this algorithm.

   If the memory gets filled, some parts of the memory will be flushed in the disk based on the flush strategy being used.

[5] AN EFFECTIVE TOKENIZATION ALGORITHM FOR INFORMATION RETRIEVAL SYSTEMS

➔ In the proposed algorithm, tokenization is done based on the set of training vectors which are initially provided into the algorithm to train the system. The training documents are of different knowledge domains, and are used to create vectors.

➔ The created vector helps the algorithm to process the input documents. The tokenization on documents is performed with respect to the vectors, use of vectors in pre tokenization helps to make the whole tokenization process more precise and successful.

➔ Tokenization involves pre-processing of documents and generating its respective tokens which is the basis of these tokens. Probabilistic IR generates its scoring and gives reduced search space.

Input: "This is an Information retrieval model and it is widely used in the data mining application areas. In our project there is a team of 2 members. Our project name is IR".

After applying tokenization process then the output is formed like:

Output: **Words**= this<1> is<4> an<1> Information<1> retrieval<1> model<1> and<1> it<1> widely<1>used<1> in<2> the<1> data<1> mining<1> application<!> areas<1> our<2> project<2> there<1>team<1> of<1> members<1> name<1> IR<1>

Numbers= 2<1>

In angular braces, the value shows the frequency of a word in the given document.

PROPOSED ALGORITHM-

```
Input (Di)   , Output (Tokens)
Begin
Step 1:
Collect Input documents (Di) where i=1, 2, 3....n;
Step2:
For each input Di;
Extract Word (EWi) = Di;
// apply extract word process for all documents i=1, 2, 3...n in and extract words//
Step 3:
For each EWi;
Stop Word (SWi) =EWi;
// apply Stop word elimination process to remove all stop words like is, am, to, as, etc.
Stemming (Si) = SWi;
// It creates stems of each word, like “use” is the stem of user, using, usage etc. //
Step 4:
For each Si;
Freq_Count (WCi)= Si;
```

// for the total no. of occurrences of each Stem Si. //

Return (Si);

Step 5:

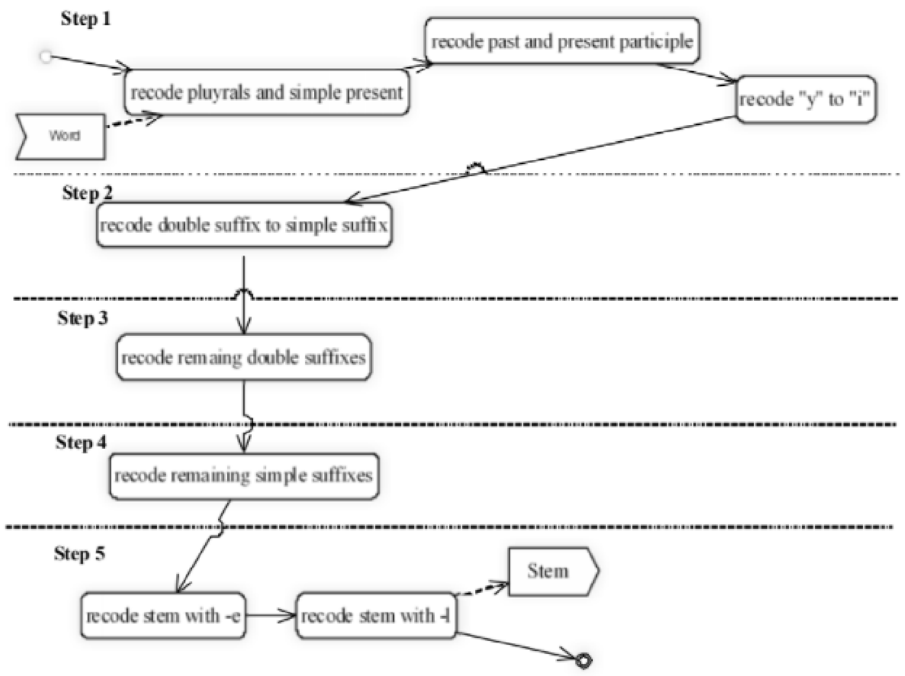
Tokens (Si) will be passed to an IR System.

End

[6] Stemming approach used for Text Processing in Information Retrieval System

A stemming is a technique used to reduce words to their root form, by removing derivational and inflectional affixes.
A word's stem is its most elementary form which may or may not have a semantic interpretation. In documents written in natural language, it is hard to retrieve relevant information.

Porter stemming algorithm is evaluated using the error counting method. With this method, the performance of a stemmer is computed by calculating the number of understemming and overstemming errors.


Step 1
Word
recode pluyrals and simple present
recode past and present participle
recode "y" to "i"
Step 2
recode double suffix to simple suffix
Step 3
recode remaing double suffixes
Step 4
recode remaining simple suffixes
Step 5
recode stem with -e
recode stem with -l
Stem

The steps of Porter Stemming Algorithm

- ➔ Specifically, the algorithm has five steps. Each step defines a set of rules.
  To stem a word, the rules are tested sequentially; if one of these rules matches the current word, then the conditions attached to that rule are tested.
  Once a rule is accepted; the suffix is removed and control moves to the next step.
- ➔ If the rule is not accepted then the next rule in the same step is tested, until either a rule from that step is accepted or there are no more rules in that step and hence the control passes to the next step.
- ➔ The stemmer is based on the idea that the suffixes in the English language are mostly built of a combination of smaller and simpler suffixes; for instance, the suffix "fullness" is composed of two suffixes "full" and "ness".
- ➔ Thus, Porter stemmer is a linear step stemmer; <u>it applies morphological rules sequentially allowing removing affixes in stages.</u> This process continues for all the five steps; in the last step the resultant stem is returned by the stemmer.
- ➔ To further confirm the robustness of the stemming algorithm, a perspective to this work is to evaluate the New Porter stemmer in other **Natural Language processing** areas.
- ➔ The New Porter stemmer will be used in order to improve automatic text summarization, <u>document clustering</u>, information extraction, <u>document indexing</u>, Question- Answering systems, etc. The new algorithm can be useful for words ‘normalization as well as reducing the space representation.

[7]An Evaluation of Preprocessing Techniques for Text Classification

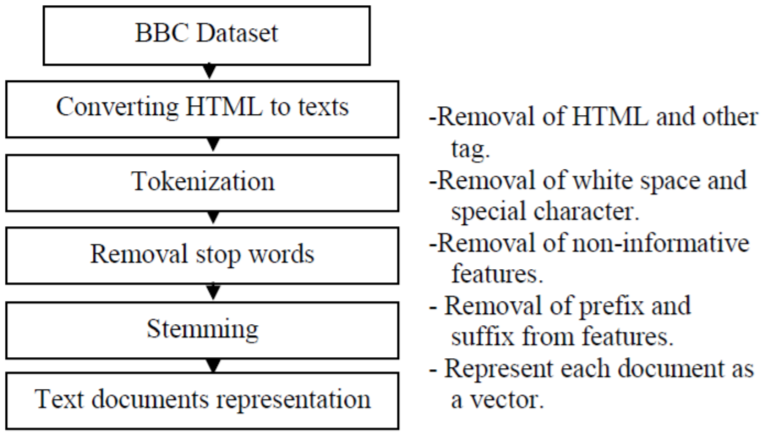
- **Text preprocessing steps**

  1) Text documents collection
  2) Tokenization
  3) Stop Words Removal
  4) Stemming

- **Indexing Techniques**

  1) Feature extraction using document frequency
  2) Feature extraction using TF-IDF (term weighting)
  3)Cosine Similarity Measure

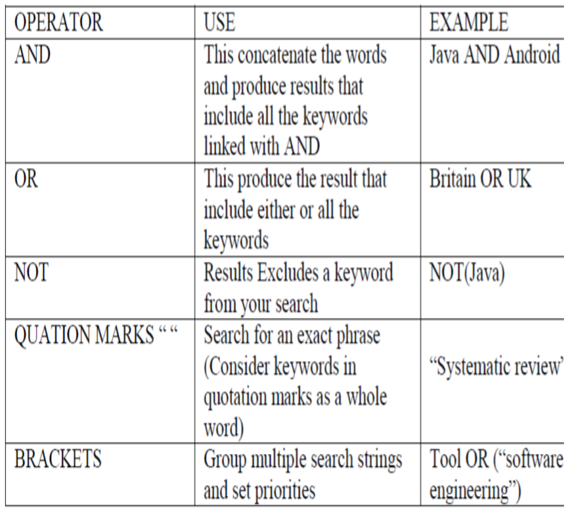
TF-IDF with cosine similarity score performed better in classifying the ten general categories based on evaluation metrics.


BBC Dataset
Converting HTML to texts
Tokenization
Removal stop words
Stemming
Text documents representation
-Removal of HTML and other tag.
-Removal of white space and special character.
-Removal of non-informative features.
- Removal of prefix and suffix from features.
- Represent each document as a vector.

[8]Efficiency of Boolean Search strings for Information Retrieval

- This model for information generates a query which is in the form of a Boolean expression of terms.
- It is the process by which the research question is translated into the research string that can be used to retrieve relevant Information.
- Taking over the control of which documents are relevant, the search for the articles is more efficient and time economical.

Source	Number of documents	Time (sec)	Method
Google scholar	3220,000	0.20	Free text query result
Google scholar	84000	0.16	Boolean search strings result



OPERATOR	USE	EXAMPLE
AND	This concatenate the words and produce results that include all the keywords linked with AND	Java AND Android
OR	This produce the result that include either or all the keywords	Britain OR UK
NOT	Results Excludes a keyword from your search	NOT(Java)
QUATION MARKS " "	Search for an exact phrase (Consider keywords in quotation marks as a whole word)	"Systematic review'
BRACKETS	Group multiple search strings and set priorities	Tool OR ("software engineering")

Irrelevant search % =(3220,000-84000)/3220,000 =97.4%

[9]Optimization of Information Retrieval Using Evolutionary Computation: A Survey

Optimization of information retrieval is done to find out the best options among a number of choices.It is being used in search engines which overall help to improve efficiency of the task.

Traditional techniques are used for the process of retrieving information but are not sufficient to optimize the retrieval process.

Genetic Algorithm (GAs) is a prevailing searching mechanisms that incorporate robustness and fast search capabilities it is suitable for retrieving information as they represent a high dimensional search space

Ant Colony Optimization (ACO) is a probabilistic technique that is helpful in problems for finding better paths by using graphs

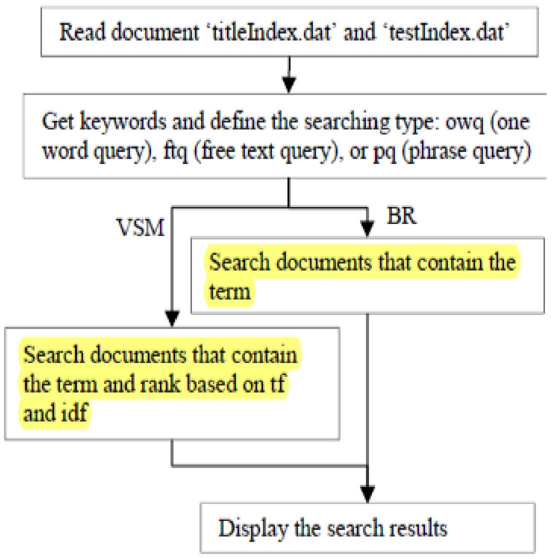
Differential Evolution (DE) is a stochastic optimization method, an improved version of Genetic Algorithm which minimizes an objective function. Real valued numerical optimization problems are solved by using few control parameters

The evolutionary computation algorithms and the process of optimization are used that help in reaching near global solutions within the search space and to retrieve better results by increasing the overall effectiveness of the system.

Base Work

[10]The Performance of Boolean Retrieval and Vector Space Model in Textual Information Retrieval

- BR method does not rank the number of terms appearing in a document because it only finds the term whether it exists or not (boolean) in documents
- The use of BR method generates an index file size smaller than the collection of source files, while the VSM method generates a larger index file than the collection of source files
- The use of the BR method requires less time than the VSM method.
- The time required to create the inverted index increases with increasing the file collection size. The time required by the VSM method is just slightly lower than the BR method.
- There is no significant time difference for creating inverted indexes for both method.


Read document 'titleIndex.dat' and 'testIndex.dat'
Get keywords and define the searching type: owq (one word query), ftq (free text query), or pq (phrase query)
VSM
BR
Search documents that contain the term
Search documents that contain the term and rank based on tf and idf
Display the search results

VSM Method

Term frequency (tf) is the frequency occurrence of a word
in a document.
Document Frequency(df) is defined to be the number of documents in the collection that contain a term.
Inverse Document Frequency(idf) = ln(N/df) , N is the total number of documents.
The idf of a rare term is high, whereas the idf of a frequent term is likely to be low
Rank = tf * idf.

Doc1	good	car	
Doc 2	good	bike	
Doc 3	good	car	bike

Example

Words	doc1	doc2	doc3
good	1/2	1/2	1/3
bike	0	1/2	1/3
car	1/2	0	1/3

Term Frequency

word	idf

car	ln(3/2)
bike	ln(3/2)
good	ln(3/3)=0

Inverse Document Frequency

Doc	Rank
Doc 1	0*ln(3/2)=0
Doc 2	0.5*ln(3/2)=0.2027
Doc 3	0.33*ln(3/2)=0.1338

Rank for Query bike ;Result: Doc 2

Creating tf and idf:

Step 1 :The application reads the corpus and processes the content.

Step 2: Perform tokenization,stop word removal and stemming.

Step 3:Create a vector posting list (inverted index) with tf and idf.

Query:

Step 1 and 2 are the same as previous.

Step 3: Create vector with keywords of query.

Step 4: Calculate the rank of each document.

Step 5: Sort the Vector in Descending order . The highest rank is the most relative document for a query.

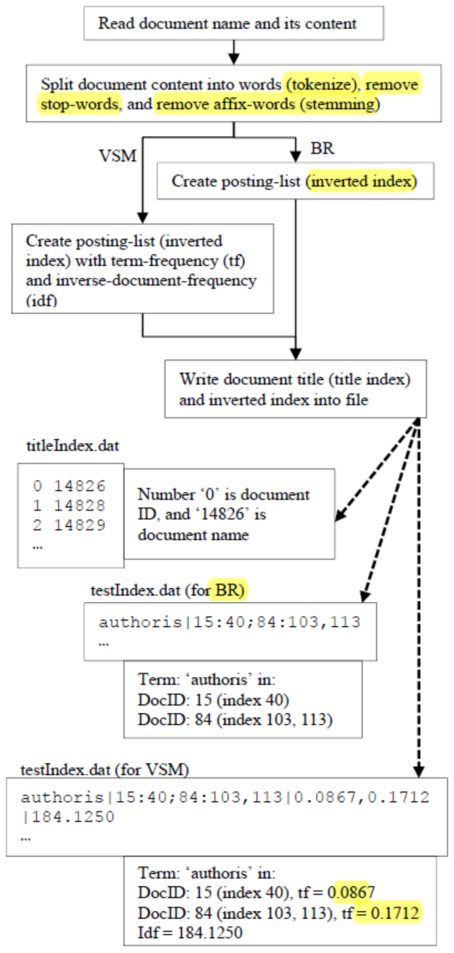
Example of tf and idf: authoris|15:40;84:103,113|0.0867,0.1712|184.1250

Term: ‘authoris’ in:

DocID: 15 (index 40), tf = 0.0867

DocID: 84 (index 103, 113), tf = 0.1712

Idf = 184.1250

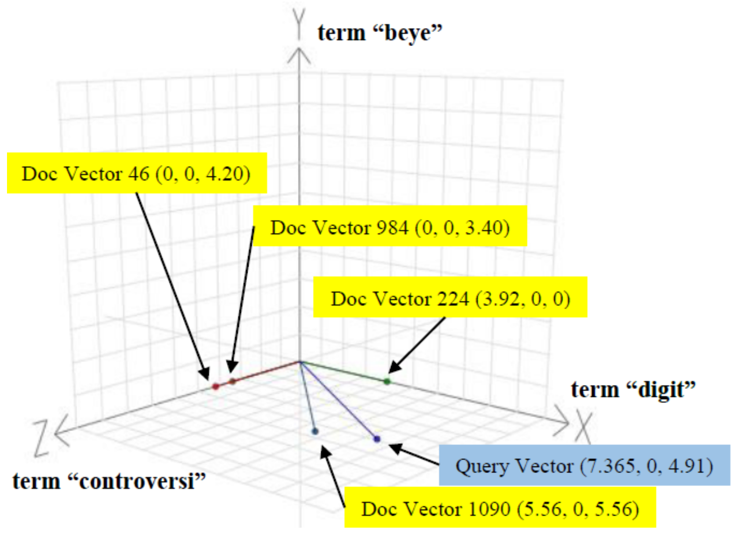

Read document name and its content
Split document content into words (tokenize), remove stop-words, and remove affix-words (stemming)
VSM
BR
Create posting-list (inverted index)
Create posting-list (inverted index) with term-frequency (tf) and inverse-document-frequency (idf)
Write document title (title index) and inverted index into file
titleIndex.dat
0 14826
1 14828
2 14829
...
Number ‘0’ is document ID, and ‘14826’ is document name
testIndex.dat (for BR)
authoris|15:40;84:103,113
...
Term: ‘authoris’ in:
DocID: 15 (index 40)
DocID: 84 (index 103, 113)
testIndex.dat (for VSM)
authoris|15:40;84:103,113|0.0867,0.1712
|184.1250
...
Term: ‘authoris’ in:
DocID: 15 (index 40), tf = 0.0867
DocID: 84 (index 103, 113), tf = 0.1712
Idf = 184.1250

Flowchart of Creating Inverted Index.

3D Cartesian Representation

The doc vector(1090) is closest to the query vector so that will be the result for the query.

Vector	Value Before Adjustment			Multiplication Factor	Value After Adjustment		
	digit	beye	controversi		digit	beye	controversi
Query Vector	$7.365 \times 10^{+2}$	0.0	$4.91 \times 10^{+2}$	1.0×10^{-1}	7.365	0.0	4.91
Doc Vector (46)	0.0	0.0	4.20×10^{-2}	$1.0 \times 10^{+2}$	0.0	0.0	4.20
Doc Vector (224)	3.92×10^{-2}	0.0	0.0	$1.0 \times 10^{+2}$	3.92	0.0	0.0
Doc Vector (984)	0.0	0.0	3.40×10^{-2}	$1.0 \times 10^{+2}$	0.0	0.0	3.40
Doc Vector (1090)	5.56×10^{-2}	0.0	5.56×10^{-2}	$1.0 \times 10^{+2}$	5.56	0.0	5.56

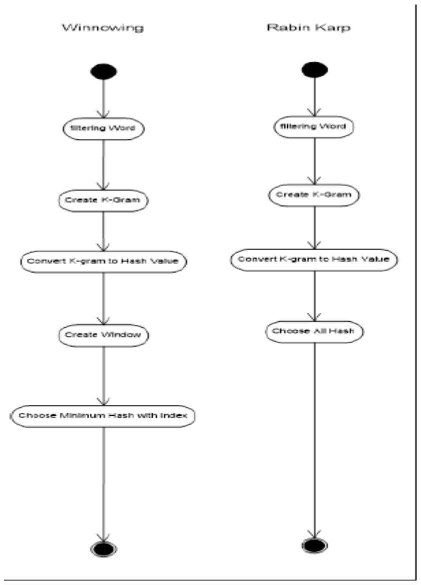

Y
term "beye"
Doc Vector 46 (0, 0, 4.20)
Doc Vector 984 (0, 0, 3.40)
Doc Vector 224 (3.92, 0, 0)
term "digit"
X
Z
term "controversi"
Query Vector (7.365, 0, 4.91)
Doc Vector 1090 (5.56, 0, 5.56)

[11] Detecting Documents Plagiarism using Winnowing Algorithm and K-Gram Method

Rabin Karp Algorithm: The Rabin Karp's algorithm computes every k-gram of the text to be equated. It also computes the hash value for the pattern. If the hash values are different, the next hash value of k-gram will be calculated. But, if the hash value is similar, then a comparison between pattern and k-gram is performed.In Rabin karp brute force is only required when hash values are the same.

Winnowing Algorithm: The algorithm input is the text document which is processed and results in output in the form of hash value. That hash value is then called the fingerprint. This is the fingerprint which is used to compare the similarity of each document.The rightmost hash value is selected if there is more than one hash with a minimum

K-gram: It is a process that converts string into substring gram. K-Gram is a substring series that is along with length k.



[12] Improving Native Language Identification with TF-IDF Weighting

- **Native Language Identification (NLI)** is the task of automatically identifying the native language of a writer based on the writer's foreign language production.
- The dataset consists of 12,100 English essays (about 300 to 400 words long) from the **Test of English as a Foreign Language (TOEFL)**
- **Features considered** - Word n-grams, POS n-grams, Character n-grams, Spelling errors.
- Term Frequency refers to the number of times a particular term appears in an essay. In our experiments, terms are n-grams of characters, words, parts of-speech tags or any combination of them.
- Inverse Document Frequency quantifies that the term which occurs in all the documents are less likely to be searched and therefore should be assigned less weight.
- **Classifiers considered** - linear support vector machines, logistic regression and perceptrons.
- Ten cross fold validation was done for each classifier.The system showed best accuracy with TF-IDF weighting.
- The feature combination that gives the best accuracy is the TF-IDF of unigrams and bigrams of words.

[13] A Survey of Concept-based Information Retrieval Tools on the Web

- **Concept-based Information Retrieval Models :** There exists the presumption that the meaning of a text (word) depends on conceptual relationships to objects in the world rather than to linguistic or contextual relations found in texts or dictionaries.Sets of words, names, noun-phrases, terms, etc. will be mapped to the concepts they encode.

- **Types of conceptual structures :**

  Conceptual taxonomy - Conceptual taxonomy is a hierarchical organisation of concept descriptions according to generalisation relationship.

  Formal or domain ontology - Ontology is a conceptual representation of the entities, events, and their relationships that compose a specific domain.

  Semantic linguistic network of concepts - NLP is used for creation of conceptual structure in some form of semantic network.

  Thesaurus - Thesaurus is a collection of words or phrases linked through a set of relationships including synonymy, antonymy, and "isa" relationship.

- **Features of Conceptual Structures :**Type of a conceptual structure, Form of representation of a conceptual structure: tree, semantic network, context vectors, conceptual graphs, rule-based language, and logic language,etc. Relationships supported by a conceptual structure: subsumption, a kind-of, a part of, associations, and relations, etc.Way of creation of a conceptual structure: manual creation, automatic learning, and NLP.

[14] Survey Paper on Information Retrieval Algorithms and Personalized Information Retrieval Concept

In the following the user will get the data that will be precise & within his/her area of interest. He needs to specify the domain and purpose of his search. It gives search results as per the requirement of the user, sending a query.

During processing of a Query. The system will know the searching domain of the user and then provides as per his requirement.

The data received is precise and can be recalled. This will help the user in getting the best result for his query, saving his time of searching & checking several non-required documents.

<u>1 - Boolean algorithm</u>

The approach of the Boolean model is as follows: Suppose, Document (D) =Logical conjunction of keywords. Query (Q) =Boolean expression of keywords and record, R (D, Q) = D®Q

$$D = t1 \cup t2 \cup \ldots\ldots \cup tn$$

$$Q = (t1 \cup t2) \cup (t3 \cup \emptyset t4)$$

D ® Q, thus R(D,Q) = 1, [1]

<u>2 - Ranking algorithm</u>

Since Boolean does not have a ranking mechanism, it may skip important data, so there was a need for ranking. The result is ranked on the basis of occurrence of terms in the queries.

This method eliminates the often-wrong Boolean syntax used by the end-users, and provides some results even though a term of the query is incorrect.

3- Vector based model

Vector space model is an algebraic document that uses vectors for representation. Documents and queries both are vectors.

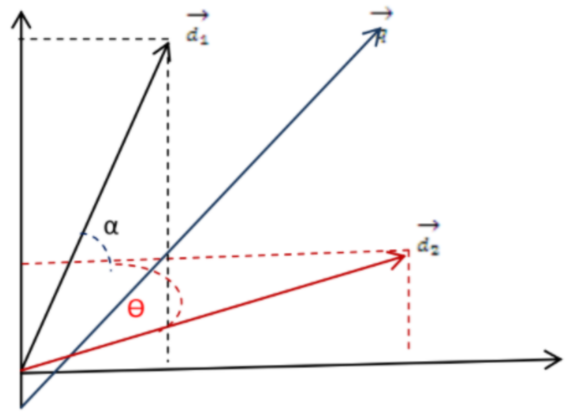


Fig1.Representation of different vectors

Where, d=document and q=query.

$$\mathrm{sim}(d_j, q) = \frac{d_j \cdot q}{\|d_j\| \cdot \|q\|} = \frac{\sum_{i=1}^{N} w_{i,j} w_{i,q}}{\sqrt{\sum_{i=1}^{N} w^2_{i,j}}\sqrt{\sum_{i=1}^{N} w^2_{i,q}}}$$

$$p_i = \frac{r_i}{R_i} q_i = \frac{n_i - r_i}{N - R_i}$$

[15] A Review on Important Aspects of Information Retrieval

The user who needs information issues a query (user query) to the retrieval system through the query operations module. The retrieval module uses the document index to retrieve those documents that contain some query terms.

The document collection is also called the text database, which is indexed by the indexer for efficient retrieval.

Indexing - It forms the core functionality of the IR process since it is the first step in IR and assists in efficient information retrieval. Indexing reduces the documents to the informative terms contained in them. It provides a mapping from the terms to the respective documents containing them.

Rank Retrieval- When the user gives a query, the index is consulted to get the documents most relevant to the query. The relevant documents are then ranked according to their degree of relevance, importance etc

Similarity Measure- Cosine , Jaccard Coefficient, Okapi

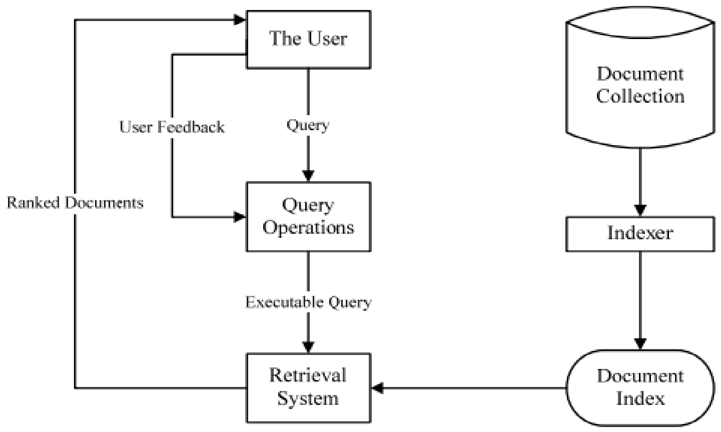
Evaluation of performance of Information Retrieval System

(1) Precision:

(2) Recall:

(3) Precision-Recall Curve:

(4) F-score:



Summary

Year and Title	Method	Drawback	Our approach
[1]M.François Sy, S.Ranwez, J.Montmain,(2015)"User centered and ontology based information Retrieval system for life sciences", BMC Bioinformatics.	Indexing techniques: Signature file, Inversion indices	Signature file based indexing is slow.	Use of Inversion indices for fast retrieval.
[2]D. C. J. Carthy, A. Drummond, J. Dunnion, and J. Sheppard, (2003)"The use of data mining in the design and implementation of an incident report retrieval system," in Systems and Information Engineering Design Symposium, Charlottesville".	Set theoretic models and Algebraic models	Difficulty in expressing complex query requirements in set theoretic models.	Use of Algebraic models which uses TF-IDF which improves retrieval performances.
[3]Bafna, P., Pramod, D., and Vaidya, A. (2016)."Document clustering: TF-IDF approach," International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, 2016, pp. 61-66	TF-IDF algorithm	Pre-processing of data should be improved.	Use of improved form of TF-IDF

[4]Nurazzah Abd Rahman & Salahi Saad Faculty of Information Technology & Quantitative Sciences Universiti Teknologi MARA, Shah Alam, Selangor,Malaysia (2004) “Hash Join Algorithms Used in Text-Based Information Retrieval: Guidelines for Users”	-Join Operation -XJoin Algorithm -Early Hash Join (EHJ) algorithm -Hash Merge Join (MHJ) algorithm	Do not have a specialized knowledge on the key values to compute a hash function	Using Enhanced Hashing Algorithms which will increase overall execution time, response time and input/output operations
[5]Vikram Singh and Balwinder Saini Department of Computer Engineering, National Institute of Technology, Kurukshetra, Haryana, India(2007) “AN EFFECTIVE TOKENIZATION ALGORITHM FOR INFORMATION RETRIEVAL SYSTEMS”	Tokenization Process and it’s analysis like -Number of Tokens Generated -Overall-Time Value - Strategy	Retrieval of relevant results is always affected by the pattern, how they are stored/ indexed which reduces it’s Efficiency	The vectors play a critical role in overall token identification and make the entire process effective and efficient.
[6] Wahiba Ben Abdessalem Karaa University of Tunis. Higher Institute of Management, Tunisia RIADI-GDL laboratory, ENSI, National School of Computer Sciences. Tunisia(2013) “A NEW STEMMER TO IMPROVE INFORMATION RETRIEVAL”	-Porter stemmer method -Comparison with Paice and Lovins stemmers	Errors made by the stemmer may affect the information retrieval performance.	The precision and the recall are improved in an information retrieval system using the new version of porter stemmer

[7]Ammar Kadhim:An Evaluation of Preprocessing Techniques for Text Classification,June 2018.	Text documents collection, Tokenization, Stop Words Removal, Stemming	Delays time execution	Effective Tokenization and improvised Stemming
[8]Muhammad Bello Aliyu:Efficiency of Boolean Search strings for Information Retrieval,American Journal of Engineering Research (AJER),2017	Boolean Search	Does not rank the number of terms appearing in a document	Usage of a model which includes ranking
[9]Shadab Irfan,Subhajit Ghosh(2017)“Optimization of Information Retrieval Using Evolutionary Computation: A Survey’’	Genetic Algorithm (GAs),Ant Colony Optimization (ACO),Differential Evolution (DE)	Very Poor Performance by traditional methods	Using evolutionary computation (EC) algorithms
[10]Budi Yulianto, Widodo Budiharto, Iman H. Kartowisastro:The Performance of Boolean Retrieval and Vector Space Model in Textual Information Retrieval,2017.	Performance Comparison	BR Model does not use frequency	Better Using VSM over BR Model
[11] Rhio Sutoyo, Insan Ramadhani, Angger Dwi Ardiatma, Sanditya Cakti Bhavana, Harco Leslie Hendric Spits Warnars, Agung Trisetyarso:(2017)“Detecting	Rabin Karp Algorithm,K-gram method, Winnowing algorithm.	In Rabin karp brute force is only required when hash values are the same.	Winnowing algorithm with k-gram method. The rightmost hash value is selected if there is more than one

Documents Plagiarism using Winnowing Algorithm and K-Gram Method”			hash with a minimum
[12]Charles S. Ahn. 2011. Automatically detecting authors’ native language. Ph.D. thesis, Monterey, California. Naval Postgraduate School.	Native Language Identification(NLI)	Accuracy is reduced even though good classifiers are used.	Use of TF-IDF weighting to improve accuracy
[13]T. Adi, O. K. Ewell and P. Adi,(1999) “High Selectivity and Accuracy with READWARE’s Automated System of Knowledge Organisation’’.	Key-based IR model and Concept based IR model	Key-based gives false search results when a word is given as input which is not in a textual object.	Use of Concept-based IR model to improve precision
[14]International Journal of Computer Applications (0975 – 8887) Volume 66– No.6, March 2013 Parul Kalra Bhatia,Tanya Mathur,Tanaya Gupta “Survey Paper on Information Retrieval Algorithms and Personalized Information Retrieval Concept”	-Boolean algorithm -Ranking algorithm -Vector based model -Probabilistic model	With growing size of database & requirement of precise data, Normal information retrieval will not give us good results and a lot of irrelevant data is also present.	Improves the performance and efficiency of the Information Retrieval Systems making it accurate and precise is what needs to be done.
[15] World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering Vol:7, No:12, 2013 “A Review on Important Aspects of Information Retrieval” Yogesh Gupta, Ashish Saini, A.K.	-Indexing -Ranked Retrieval -Development of Similarity Measure	Relative inconvenience of accessing information, Slow speeds of the internet delay the	-Whole resources found on the querying method on a centralized database. -To respond immediately when a new search query is found.

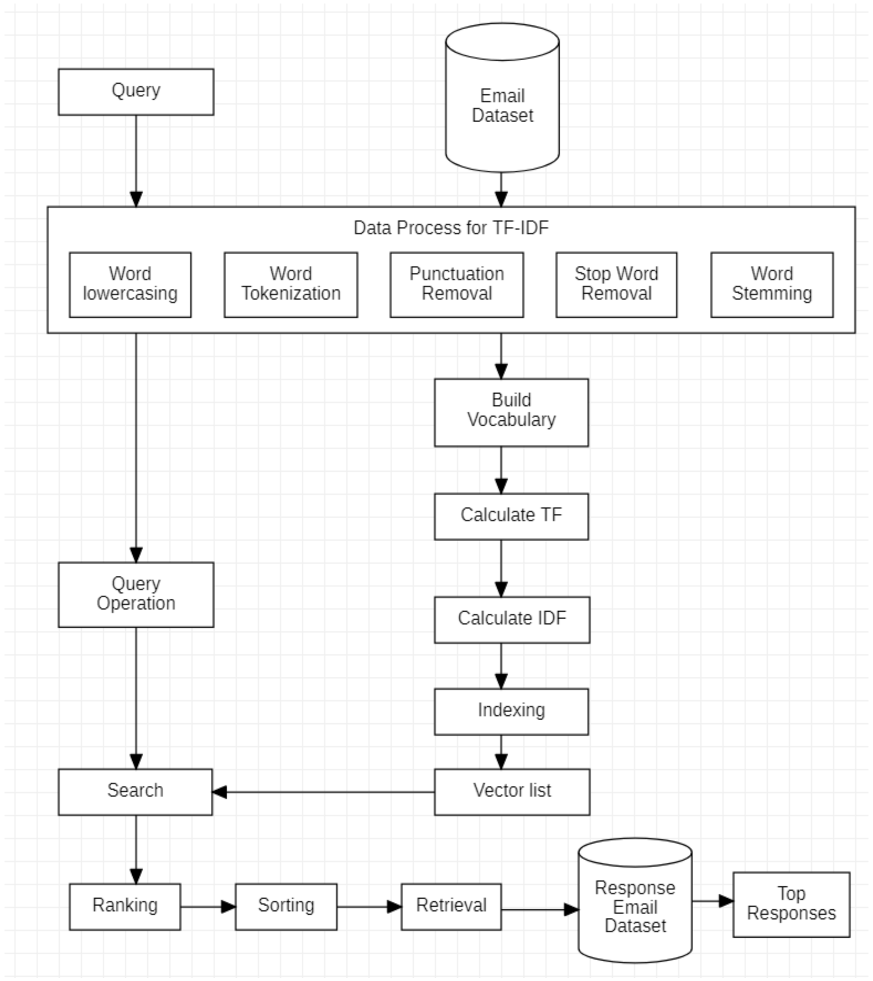
Saxena	-Emergence of Query Expansion	retrieval system.	

Comparative Analysis

Sl.No	Method	Strength	Weakness
1	Boolean Model	Flexibility and exuberance.Allow complex search requirements.	No ranking of retrieved documents
2	Vector Space Model (VSM)	Relevance scoring and relevance feedback.	Implementation problem and vector components values undefined.Computational expense.
3	Probabilistic Model	Used to Retrieve documents based on probability of relevance. Operates recursively.	Hard to build and program.Complexity grows quickly.
4	2-poisson Model	Not Require an additional term parameters Weighting algorithm.	Problem in estimation of the parameters. Require probabilities conditioned on relevance.
5	Network Model	Include terms dependence of relationships.Diminish computation cost by using simplifications	As query nodes increase, use of the network is impractical. Node growth is exponential with a number of parents.

6	Bayesian Model	Analyze Complex assumptions	Time required assessing the distribution and space requirement is NP hard problem.
7	Language Model	Novel way of Looking for a problem. Probabilistic language modeling	No notion of relevance in the model: everything is random ,query expansion is not part of the model, does not directly allow weighted or structured queries.
8	Google page rank model	Less Time Consuming,Computes rank score at indexing time not at query time	Favours older pages. Rank sinks problem. Dead Ends. Reduce front Page's Page Rank if circular reference. time. Spider traps if no links from within the group to outside the group. Dangling links problem.

High level Design



EXPLANATION:

- Email dataset consists of all the email files considered.
- Data Processing happens for all the documents which include lowercasing of words, tokenization, stemming, stop word removal and finally removal of punctuations.
- The term frequency(TF) is calculated for all the words after data processing.
- Once the TF is calculated, the inverse document frequency(IDF) is calculated and indexing is done on the words and is stored as a vector.
- Once the query is given, the data processing happens on these words too.
- Next, the search happens in the vector list and ranking is calculated by TF*IDF.
- Then the sorting is done in descending order and the retrieved emails will be stored in response email dataset.
- Among them, the top 10 responses are shown.

Functional and Non functional Requirements

Functional Requirements :

- A model of Documents
- A set of valid queries
- Retrieving valid indexes.
- Preprocessing of dataset without losing valuable information.

Non Functional Requirements :

- The search will take very less time satisfying the user's needs.
- The search will be reliable to all queries provided the query exists in documents.
- The search is more efficient and fast when the query includes boolean expression of terms.
- Every unsuccessful attempt by a user to access information of data shall be recorded on an audit trail.
- The algorithm should be capable enough to handle huge dataset without affecting its performance

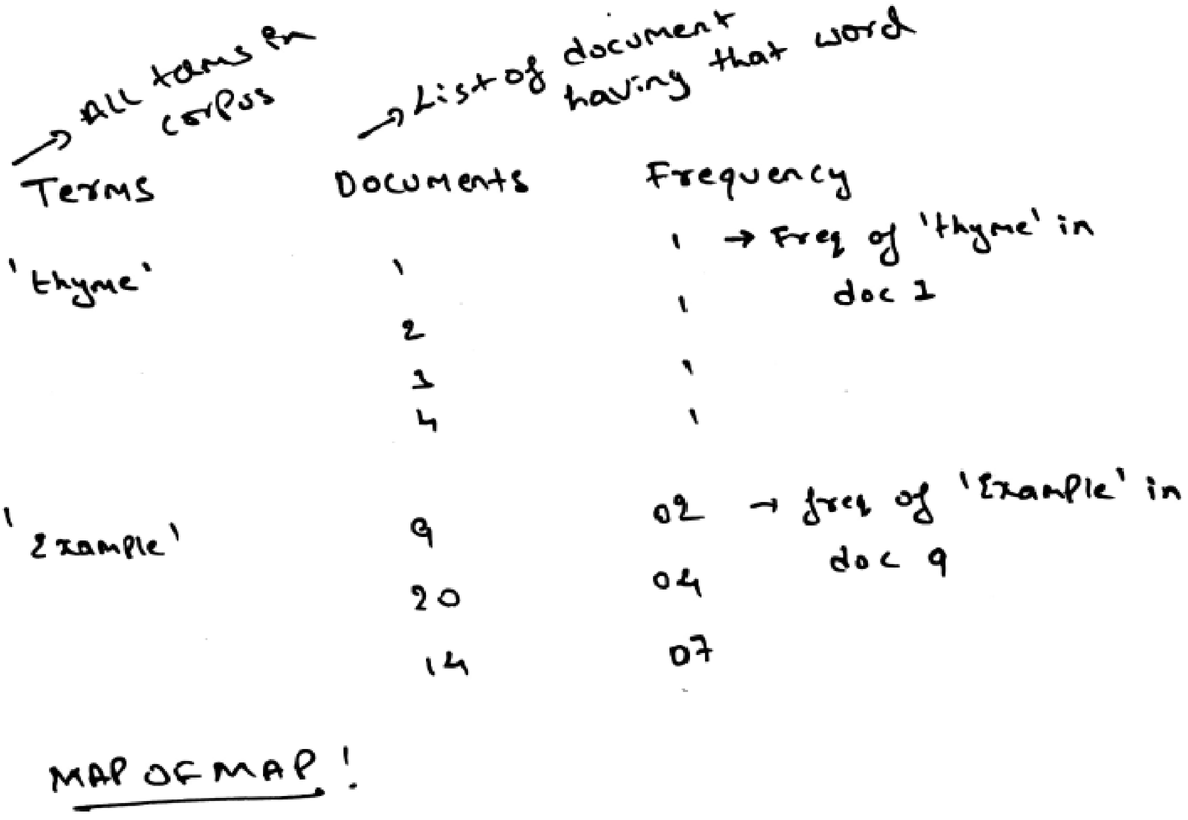
Implementation

## Importing Libraries

```
import pandas as pd
import numpy as np
import nltk    #nltk Library used for Data preprocessing
from nltk import word_tokenize # Used for Tokenization of file
import pathlib # extract the file from desktop
from nltk.stem import PorterStemmer # Stem the file
from nltk.corpus import stopwords  # Stopword removal
from nltk.tokenize import word_tokenize
import math


 ## Extracting English StopWords from stopwords

from nltk.corpus import stopwords
stopwords=stopwords.words('English')
print(stopwords)

## PreProcessing a Sample Files

## Importing files from folder name 'sample'

file=[]
for path in pathlib.Path("D://desktop//sample").iterdir():
    if path.is_file():
        current_file = open(path, "r")
        file.append(current_file.read())
        current_file.close()

## Tokenization of files

token_file = []
for x in file:                              #select one file at a time from the folder and tokenize and
append to token_file[]
    token_file.append(word_tokenize(x))
token_file[0]



## Stemming of files
```

```
stem_file = []
ps = PorterStemmer()
for x in token_file:                    #Select one file at a time from token_file[] and perform
stemming and append to stem_file[]
    temp=[]
    for i in x:
        temp.append(ps.stem(i))

    stem_file.append(temp)


## Removal of stop words from stemmed file

stems_without_sw = []
for x in stem_file:
    stems_without_sw .append([word for word in x if not word in stopwords])


## Removal of Punctuation from the files

stems_without_sw_punct = []
for x in stems_without_sw:
    stems_without_sw_punct .append( [word for word in x if word.isalnum()])
stems_without_sw_punct


## Applying TF-IDF On Corpus
## Importing all the file paths

import os
file_names = []
for path, direc, files in os.walk("corpus"):
    for i in files:
        file_names.append(path+"/"+i)
num_of_doc=float(len(file_names))
```

```
## Applying Preprocessing and Term Freq for Each file.

length_of_docs={}
term ={}
ps = PorterStemmer()


for x in file_names:                                #One file at a time from the corpus
   file_content = open(x).read()
   token_file=(word_tokenize(file_content))             #Applying Tokenization
  # print(token_file)

   stem_file = []                                #Applying Stemming
   for i in token_file:
      stem_file.append(ps.stem(i))


   stems_without_sw = ([word for word in stem_file if not word in stopwords])
#StopWord Removal

   stems_without_sw_punct = ( [word for word in stems_without_sw if word.isalnum()])
#Punctuation Removal

   pre_file=stems_without_sw_punct

   #print(pre_file)
   length_of_docs[x]=len(pre_file)

   for i in pre_file:                          # for each word in mail storing document it is
present
                                          # and also the frequency of the term
      if i not in term:
         term[i]={}
         term[i][x]=1
      else:
         if x in term[i]:
            term[i][x] += 1
         else:
            term[i][x]=1
```



→ All terms in corpus

→ List of document having that word

Terms	Documents	Frequency
'thyme'	1	1 → Freq of 'thyme' in doc 1
	2	1
	3	1
	4	1
'Example'	9	02 → freq of 'Example' in doc 9
	20	04
	14	07

MAP OF MAP!

```
## Cal Document Frequency
doc_freq={}

for i in term:
   doc_freq[i]= ( len(term[i]) )


## Applying TF-IDF value for each term
for i in term:
   for j in term[i]:
        (term[i])[j]=(math.log10(1+(term[i])[j])) * math.log10(num_of_doc/doc_freq[i])



## Query Process
## Input Query and PreProcessing

query = input("Enter your query: ")

token_query=(word_tokenize(query))
stem_query = []

for i in token_query:
        stem_query.append(ps.stem(i))

query_stems_without_sw = ([word for word in stem_query if not word in stopwords])

query_stems_without_sw_punct = ( [word for word in query_stems_without_sw if
word.isalnum()])

pre_query_file=query_stems_without_sw_punct

pre_query_file



## Cal of term frequency of input query
 query_term={}

for i in pre_query_file:
    if i not in query_term:
        query_term[i] = 1
    else:
        query_term[i] += 1


## Normalization of Term Freq
for i in query_term:
    query_term[i]=math.log10(1+query_term[i])
```

```
## Cal tf-idf value for each document consisting query words

retrieve = {}

for i in query_term:
    if i in term:
        for j in term[i]:
            if j not in retrieve:
                retrieve[j] = query_term[i]*term[i][j]
            else:
                retrieve[j] += query_term[i]*term[i][j]


## Sorting Based on TF-IDF Value
result = []

result= sorted(retrieve.items(), key=lambda x: x[1],  reverse=True)
#result

## Retrieving Top 10 Relevant Mails For The Input Query

i=0
while i<len(result) and i<10:
    print(result[i])
    i += 1
```

Testing

Unit Testing on each Pre-Process step

```
## PreProcessing a Sample Files

## Importing files from folder name 'sample'

file=[]
for path in pathlib.Path("D://desktop//sample").iterdir():
    if path.is_file():
        current_file = open(path, "r")
        file.append(current_file.read())
        current_file.close()

## Tokenization of files

token_file = []
for x in file:                          #select one file at a time from the folder and tokenize and
append to token_file[]
    token_file.append(word_tokenize(x))
token_file[0]



## Stemming of files

stem_file = []
ps = PorterStemmer()
for x in token_file:                    #Select one file at a time from token_file[] and perform
stemming and append to stem_file[]
    temp=[]
    for i in x:
        temp.append(ps.stem(i))

    stem_file.append(temp)




## Removal of stop words from stemmed file

stems_without_sw = []
for x in stem_file:
    stems_without_sw .append([word for word in x if not word in stopwords])
```

```
## Removal of Punctuation from the files

stems_without_sw_punct = []
for x in stems_without_sw:
   stems_without_sw_punct .append( [word for word in x if word.isalnum()])
stems_without_sw_punct
```

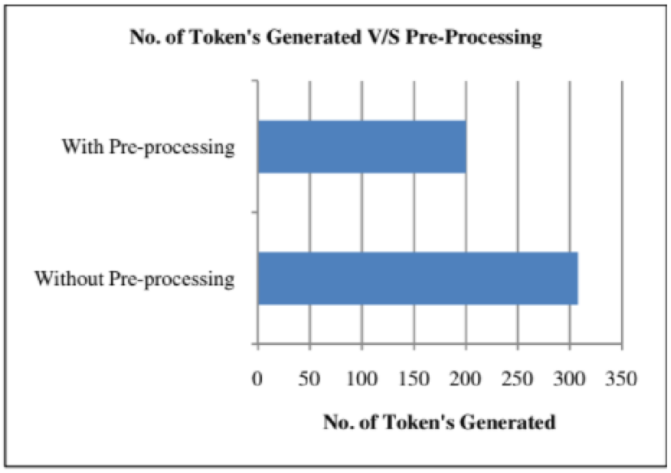
Explanation:

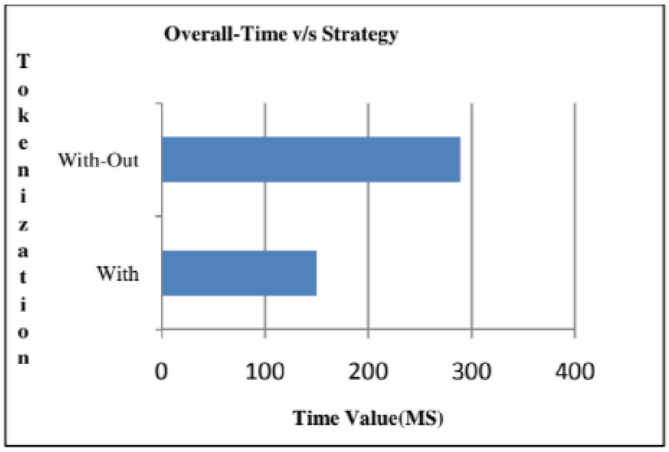
- Unit Testing is done on a sample file which consists of 11 files.
- First step is to read the contents from the files which are present in the path provided.
- Next step is to tokenize the words from the contents of the file read.
- Once the tokenization is complete, stemming is performed on the tokenized document.
- Next step is the removal of Stop words from the stemmed file
- Once all this is complete, the last step is to remove punctuations from the files.
- This completes the preprocess of the sample file and the file will be ready for TF-IDF.

Results and analysis - Tokenization Process

For one of the research paper we have analysed based on certain parameters <u>(1) Number of Tokens Generated (2) Strategy (3) Overall-Time Value</u>

The results shown in the paper are based on the experimentation over more than 100 input documents and more than 50 input document vectors. Further, for the comparative analysis below mentioned parameters are used:





<u>Analysis of Various Hashing Algorithms used in IR system</u>

<u>1.Execution Time</u>

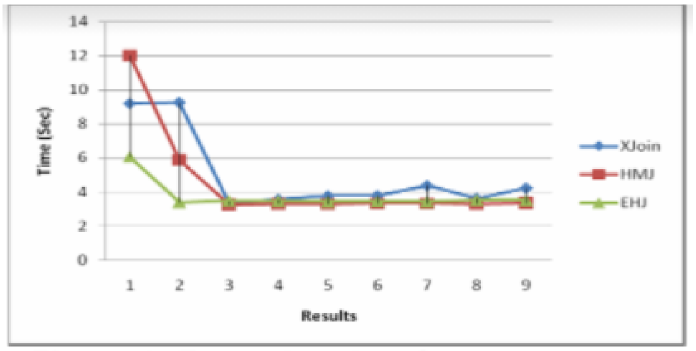


Figure 1: Execution time for *one-to-many* hash join type

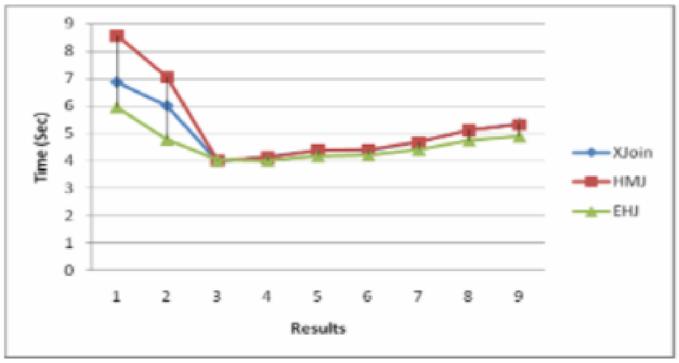


Figure 2: Execution time for many-to-many join type

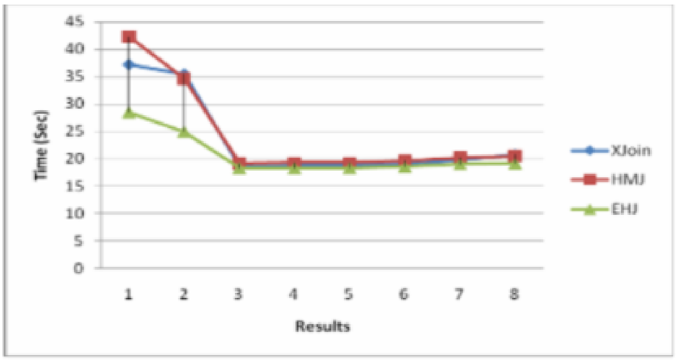


Figure 3: Execution time for multi-relations join type

2. Input-Output Operations Performed

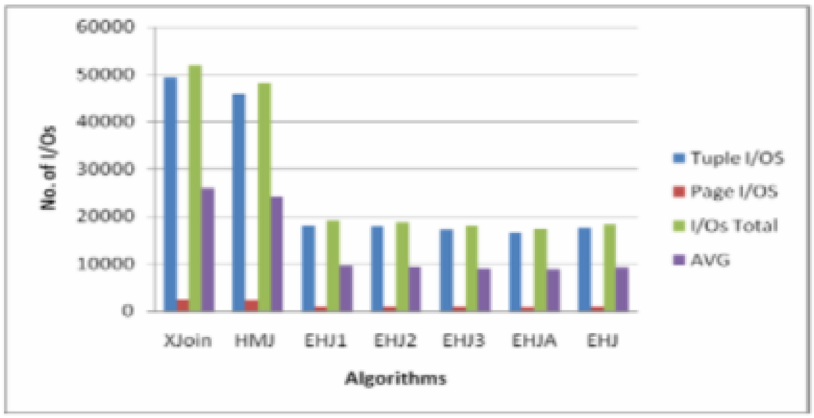


Figure 4: I/Os performed for one-to-many join type

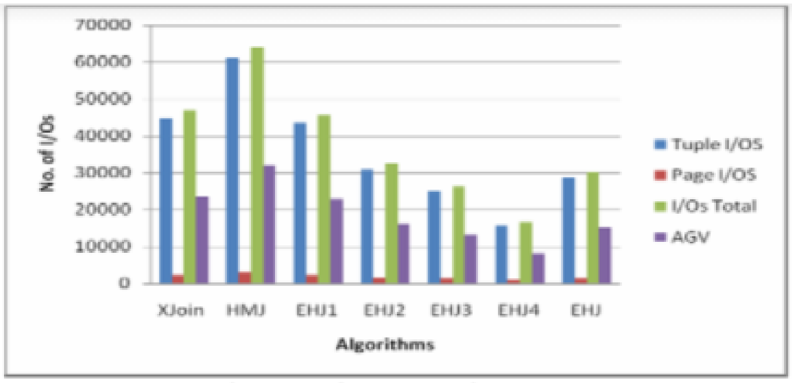


Figure 5: I/Os performed for many-to-many join type

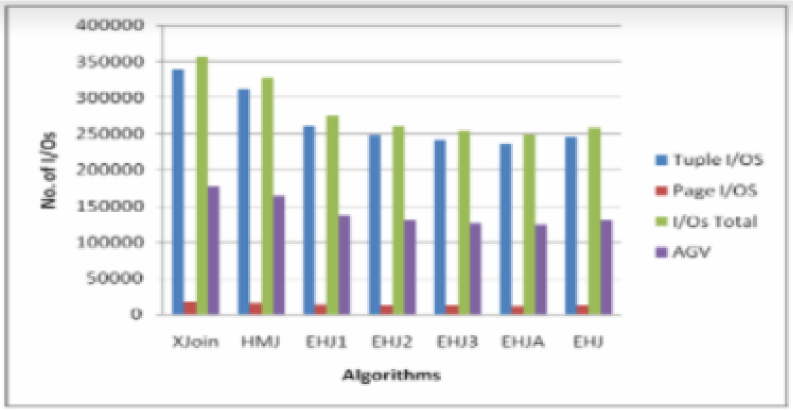


Figure 6: I/Os performed for multi-relation join type

Conclusion

- The corpus is a collection of thousands of mails from the kaggle website.
- The main objective of retrieving top 10 relevant mails based on the user query from the corpus is implemented.
- Various pre-processing methods and algorithms like hash join , effective tokenization , stemming , stop word and punctuation removal were analysed and executed which transforms the unstructured data to build the model achieving better performance.
- The execution time and performance of the TF-IDF model without preprocessing are high and poor respectively.
- Identified VSM model as best suitable for the objective among the different IR models and implemented VSM model on training corpus and tested against user query. Showed good results when checked manually.
- Porter's algorithm is used for stemming. Since it's a basic stemming technique, other stemming methods like porter2 and etc can be used to improve the performance.
- Concise queries are very much required in order to satisfy user’s needs based on the searched query. Different types of hashing algorithms are used in IR which are varied based on it’s techniques involved in join operations. Analysis of algorithms based on Overall Execution Time , Response Time , I/O Operations has been done.

Future Work

- Designing a responsive and interactive user interface which takes the query and displays relevant mails , functions and depicts the whole working of VSM Model.

- Using other different models like probabilistic models, algebraic, logical models, information theoretic models, and Bayesian models which selects and ranks the relevant documents with respect to a user's query.

- TF-IDF model can not only be used just for retrieval of Emails , it can be used for other domains like Media search , Blog search , Information filtering , Recommender systems and is used by search engines as a central tool in scoring and ranking a document's relevance given a user query.

- With the advent of the World-Wide Web and the huge increase in the use of the Internet , different retrieval methods like retrieving any word or line of the query in the document can be used to index Web pages and provide access to them.

- Evaluating using different performance metrics to validate the TF-IDF model and also on other models and comparison between them.

- Advance pre-processing techniques like hash join,effective tokenization and other nlp libraries like spacy or textblob etc to increase the performance of the model and fasten the results.

REFERENCES

[1]M.François Sy, S.Ranwez, J.Montmain,(2015)“User centered and ontology based information Retrieval system for life sciences”, BMC Bioinformatics.

[2]D. C. J. Carthy, A. Drummond, J. Dunnion, and J. Sheppard, (2003)“The use of data mining in the design and implementation of an incident report retrieval system," in Systems and Information Engineering Design Symposium, Charlottesville”.

[3]Bafna, P., Pramod, D., and Vaidya, A. (2016)."Document clustering: TF-IDF approach," International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, 2016, pp. 61-66

[4] G.H. Gonnet, R. Baeza-Yates, (1991.)“Hash Join Algorithms Used in Text-Based Information Retrieval”

[5] G. Salton, M.J. Mcgill,(1983) “Introduction to Modern Information Retrieval”, Mcgraw-Hill Book Co., New York,

[6]] Lovins, J.B. (1968). Development of a stemming algorithm. Journal of Mechanical Translation and Computational Linguistics, Vol. 11,No. 1 and 2, pp. 22-31.

[7]Ammar Kadhim(June 2018)“An Evaluation of Preprocessing Techniques for Text Classification”.

[8]Muhammad Bello Aliyu (2017)“Efficiency of Boolean Search strings for Information Retrieval,American Journal of Engineering Research (AJER)”

[9]Shadab Irfan,Subhajit Ghosh(2017)“Optimization of Information Retrieval Using Evolutionary Computation: A Survey”

[10]Budi Yulianto, Widodo Budiharto, Iman H. Kartowisastro(2017)“The Performance of Boolean Retrieval and Vector Space Model in Textual Information Retrieval”.

[11] Rhio Sutoyo, Insan Ramadhani, Angger Dwi Ardiatma, Sanditya Cakti Bavana,

Harco Leslie Hendric Spits Warnars, Agung Trisetyarso:(2017)“Detecting Documents Plagiarism using Winnowing Algorithm and K-Gram Method”.

[12]Charles S. Ahn. (2011). Automatically detecting authors’ native language. Ph.D. thesis, Monterey, California. Naval Postgraduate School.

[13]T. Adi, O. K. Ewell and P. Adi,(1999) “High Selectivity and Accuracy with READWARE’s Automated System of Knowledge Organisation’’.

[14] Norbert Fuhr “Probabilistic Models in Information Retrieval”. Brown, L. D., Hua, H., and Gao, C. (2003). A widget framework for augmented interaction in SCAPE.

[15] Singhal A (2001) Modern Information Retrieval: A Brief Overview. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering.