ISSN: 2633-4828 Vol. 5 No.4, December, 2023

International Journal of Applied Engineering & Technology

INTELLIGENT WORD AND PHRASE ANALYSIS AND PREDICTION TOOL

Shashi Pal Singh[1], Ritu Tiwari[2] and Sanjeev Sharma[3]
[1]AAIG, Center for development of Advanced Computing (C-DAC), Pune, India
[2,3]Indian Institute of Information Technology (IIIT) Pune, India
[1]shashis@cdac.in and [2]ritu@iiitp.ac.in

ABSTRACT
In the modern-day world global expansion of technology has increased. Access to the internet is unlimited due to internet of things which produces an increasing amount of data. This leads to the rapid growth of digital data storage. As digitalization evolves, the question arises of how to effectively manage and find the clean, quality, and reliable data. As the storage of data increases in petabytes and exabytes the file structures are organizing in such a way that the information retrieval of searched keywords matches the speed of emerging digital data storage. Consequently, this leads to the need of information retrieval from the file system. If we apply direct keywords searching techniques in the files then the obtained results will be inefficient and inconsistent. To overcome these challenges the proposed tool searches.

Index Terms – Indexing, Searching, KWIC (Keyword in context), Natural Language Processing, Prediction

INTRODUCTION
The word and phrase prediction tool forecasts the intended words and phrases to the user from the file system. It is also known as the next word prediction as it monitors the letters entered by the user and produces the list of documents that has the recorded sequence of letters entered by the user. In other words, it can be said that it is a higher-order word processing feature that takes the edge off the writing breakdowns, by reducing the number of keystrokes that are necessary for typing by the user. As digital platforms are increasing word prediction tools are becoming more and more popular. This tool can be developed by implementing keyword indexing and searching.

As technology is rapidly evolving the statistics state that a number of people using the internet has rapidly grown and, more and more people are using the technologies available. Users who have recently started using such technologies may find difficulty in finding the required information because data is available in large volumes. Also, many people may get confused by the pronunciation of the word and hence can misspell the word while writing it down because of which they may not get the results that they desire and may move on to another source to get the information.

It's a minor feat, but there are many people who type considerably slower than they read, a lot of time is consumed by them while typing their full query.

It's a minor feat, but there are many people who type considerably slower than they read, a lot of time is consumed by them while typing their full query. There are many people with disabilities such as dyslexia and dysgraphia having difficulty with reading and spelling because of which they find it difficult to write the text quickly and correctly [1].

Word and phrase prediction tools can be a solution to all such kind of problems. It predictively searches the user query and fetches the results much faster using keyword indexing, making the searching process easier for the user. As the user enters the keywords or phrases for searching the query word prediction tool provides the list of documents that has the keywords entered by the user.

This tool tries to personalize the search to an individual.

Word prediction is assistive technology as it serves as assistance to the people who lagged behind in typing or do not have good lexicon knowledge. The main purpose of this tool is to sustain or refine an individual's performance and independence to facilitate participation by increasing overall functioning. It tries to support

people with dyslexia or dysgraphia to help them write text fast and accurately. It is an assistive tool but many people also use it in their day-to-day daily life and hence valuable for writers with typing issues and phonetic-based misspellings. There are many peoples having different issues. The proposed tool tries to solve all of these problems.

This tool focuses on making typing and searching tasks easier and faster for the user to search for the terminologies related to the keyword by carrying out indexing and searching techniques in uploaded files or in unstructured data, providing minimum keystrokes.

Word and phrase prediction tool proposed here predict words and phrase on the basis of prior knowledge and reasoning. As predictive intelligence is a crucial tool for today's world, it chooses the most meaningful predictions and recommendations strategically hence, try to prevent user from opting out, encourage others to spend more time using the tool, and builds the long-term value for the system.

The quintessence of word and phrase prediction tools is to map the linguistics to provide the relevant system output corresponding to the user input using keyword indexing and searching techniques. Keywords are the terms that capture the essence of the subject of the document. The purpose of keyword searching is to make the search easier by making articles findable in the file system. To speed up the searching process keyword indexing pre-calculates the location of keywords and indexes it to the associated documents in advance. Indexing reduces the time consumed in searching of the keyword. For finding any keyword first indices of the document is created then the keyword is located by searching into the index of the document. Thus, providing word or phrase prediction faster and increasing KSS (keystrokes saving).

In the present world as we are observing people with disabilities is increasing day by day. Every 1 person in 10 million individuals have some form of disorder and such people are increasing every year accordingly. Aging and other factors are the socioeconomic trends that have contributed to the growth of a large number of the population that are categorized as disabled, making disability an important issue for developers.

The motivation is to provide a more efficient tool for the prediction of words and phrases than the existing tools which improves the user's writing speed and accuracy of the system. This tool shows promising results in decreasing spelling and typographical errors. It focuses on predicting the words based on the inverted indexing technique for information retrieval from the document in the file system. The predicted words or phrase can be displayed to the user in a different format that is selected by them according to their convenience.

LITERATURE REVIEW

The word and phrase prediction tool is a very helpful tool that can make a huge difference in the thinking and learning of people. It predicts what might come next by suggesting better terminologies and focuses on the discovery of the hidden relationships between variables, rather than just determining the like outcomes. Predicting the set of choices, it helps in reducing the keystrokes. A keystroke is an act of depressing one of the keys on a keyboard. While typing a character, a keystroke is performed every time a key is pressed. Keystroke saving is saving the number of keys pressed by used. 37%-47% keystrokes saving was found in various existing word prediction systems [2].

Word prediction is an adaptive tool that increases productivity by increasing keystroke saving. It also provides spelling assistance that greatly benefits people with learning disabilities, communication impairment or motor impairment. It provides a helping hand to slow typists for the entry of text.

To predict any word or phrase efficiently, the system has to extract the essential information which can be done by natural language processing tasks for preprocessing of data [3]. The Natural Language Processing (NPL) task transforms the entire text entered by the user and extract the meaningful information which includes linguistic expressions filtering out the common words and unnecessary data from the text under pre-processing. This avoids a retrograde effect on further operations that will be performed on data. Different NLP tasks are used to perform

various text pre-processing steps such as stop words removal, stemming, POS tagging, parsing, and information extraction.

Coreference Resolution finds words that refer to the same entity in a sentence. Discourse Analysis understands the text level-wise and correlates it with other sentences. Named Entity Recognition (NER) resolves each word present in the sentence into its component part and groups them in predefined class accordingly. Sentiment Analysis extracts the writer's perspective on a particular matter in the given data. Word Sense Disambiguation (WSD) classifies the words according to their meaning in a sentence. Stemming derives the word to its original form. Parts of Speech (POS) indicates how the word is functioning in the sentence, with help of its grammatical form. Chunking obtains important individual pieces of and groups them together. This helps in obtaining those documents in a result that contains the relevant keyword. By using these NLP tasks, we can perform text preprocessing to obtain keywords that retrieve the desired document.

In the study by B Mukherjee [4], in the indexing system, index entries are created of text documents for keyword indexing without any control over vocabulary. In this process, the index is created of each document along with the offset of the keywords where they are present. They described variants of keywords namely Keyword-With-Context, Keyword In Context, Keyword Augmented in Context, key-term alphabetical, word and author Index, double KWIC Index, key-letter-In-context, and keyword Out of Context. These are the different formats in which the output can be obtained after searching for the predictions in the database.

Speed and economy is the main merit of keyword indexing with which an index can be produced. This technique has therefore become feasible to issue keyword indexes at frequent intervals. Perfect consistency and predictability can also be achieved even if there is an absence of interpretation of contents.

Unless the word is present in the stop-wordlist, if a word appears in the title of the document then it is confirmed that an entry will be generated under that word. With human indexers ,there is no possibility of inconsistency in the allocation of terms in such systems which is why the scope for error or dispute does not exists. In the modern world ,human intellect has become costlier day-by-day and difficult to find, in the knowledge-based economy. But here it does not require intellectual effort or trained indexing staff as it is produced by a computer. Nowadays most user search for the document by title rather than its author. To give accurate results in such ca ase keyword indexing is very helpful. Now, the authors use representative and meaningful titles which summarize the contents of documents. Satisfying the current approaches of user's keyword indexing leads to better results.

Indexing is a technique of retrieving the data from the documents and search weather the keywords is found in the document or not. When we store any document in our system, we have to encode the subject of our document which means we should be able to identify the subject as well. For indexing, indexer do not have enough understanding to comprehend the meaning of the text written in the document. Therefore, it relies solely on the information which is given in the document such as title, abstract, introduction, preface, publishers of the document and retrieve information through them. It does not add data from its own or from any other source. It directly derives the data from the given document. This method for retrieving information from the documents is called as inverted indexing. Inverted indexing is used in information retrieval systems. Inverted indexing is of two types namely title-based indexing and citation indexing [5].

Title-based indexing: The title of the document tried to define the subject or content of the document itself. It is a kind of summary of the document. Hence to retrieve information from the document title is used as an index point. When using computer, it is very easy as key terms which represent the document are already present in the title. While implementing title-based indexing it is important to keep in mind that title are not always selected as to represent the subject of the document. Therefore, it is effective only if the title of the document is clearly expressed. It is also known as keyword indexing and its examples are keyword in context, key-term alphabetical and keyword out of context. The format contains keywords, context and location of a keyword.

The indexing process consist of three steps. In first step, all the common words and non-significant words are removed and important keywords are selected from the title of the document by the editor. The selection of key terms is done with the help of stop-list which contains list of non-significant words in it. When editor selects keywords, it removes those words from the title which are present in the stop-list. In second step, the editor moves the title in such a way that the keyword appears in the extreme left or in the centre of the content. In last step index entry for a document is generated and while its location in the document.

Citation Indexing: It is a well-ordered list of cited articles and citing articles. Here list of cited articles recognises as reference for the citing article which acts as source. The citation index is prepared by associating the cited and citing articles. Here whenever a new article cites already existing article then these two articles are associated together. Since cited and citing articles are intellectually related to each other that is why citation indexing tends to give better related to other indexes and it is comparatively less complex then others. It does not have semantic and vocabulary difficulties and it follows scientific disciplines.

There is various type of search for the searching process in information retrieval system indexer are provide with the list of keywords which describes the content of the document stored in file system.

***Phrase search*:** It allows user to retrieve those results which contains specific combination of words. It searches only for those results which contains the exact spelling such as "Word and Phrase Prediction Tool".

***Concept Search*:** It searches for electronically stored unstructured data which is conceptually similar to each other such as email.

***Concordance Search*:** It searches for specific source segment and retrieve the keywords with their immediate context.

Proximity Search: It searches for those term which occurs simultaneously with in a number of words and retrieve the documents accordingly.

***Fuzzy Search*:** It searches for those documents which contains the keywords in their variations.

***Wildcard Search*:** This search replaces one or more letters in the user query with character such as asterisk. Astrisk may represent any number of characters. For example, if we search for educat*, wildcard search will retrieve documents which contain educate, education, educator, and educational.

The study of Premalatha.R and Srinivasan.S on Text Processing in Information Retrieval System Using Vector Space Model shows that text processing is use for information retrieval by analysing and manipulating the textual information given in the document using vector space model is used [6].

The similarity score or rank between the words in documents and vectors is a cosine similarity score and is represented by, cosine similarity

$$\cos(LM) = \frac{\vec{L} \cdot \vec{M}}{|\vec{L}| \cdot |\vec{M}|}$$

Where D and Q are document and query vectors, respectively.

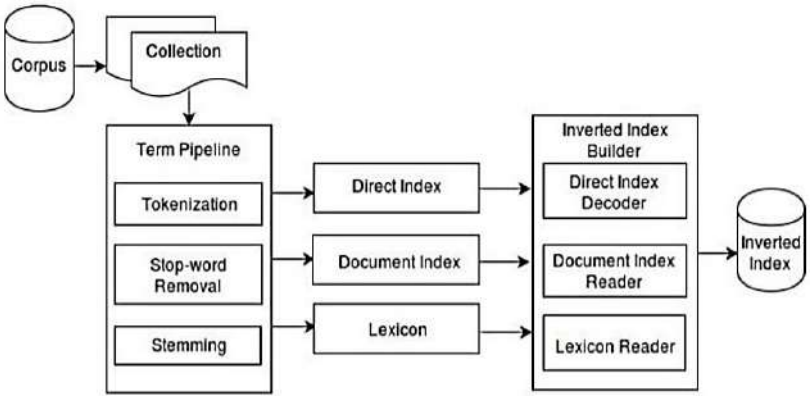


Fig.1. Indexing process

This model accesses the value of each term by assigning them numeric value according to their importance in document term weighting is used. Term frequency assign weight equal to occurrences of term(t) in document(d). Document frequency defines number of documents contain term(t). Inverse document frequency (IDF) is log(N/df) where, N is total number of documents in a collection and idft is document frequency. IDF defines importance of term in the document. Tf-idf weighting is product of tf and idft, it assigns composite weight to term(t) in document(d).

$$\text{tf-idf} = \text{tf} \times \text{idf} \quad (1)$$

$$\text{idf}(t) = \log \frac{n+1}{\text{df}(d,t)+1} + 1 \quad (2)$$

The document which has highest term weight will be suggested when input is given. Existing system use morphology for information retrieval by reading whole sentence whereas this approach is more effective as it can read a text or a character which is searched and gives suggestion accordingly.

In a study of Indexing process and evaluation [7], it is suggested that indexing plays an important role in information retrieval. To simplify the indexing process in Fig. 1, it is divided into four stages namely content specification, tokenization of documents, processing of document terms and index building. To perform indexing process on textual data different pre-processing steps are performed for content specification. These text operations are tokenization, stop-words removal and stemming. Tokenization performs syntactic analyses on the document and resolve it into component parts. It also removes punctuation marks from the sentences. Stop words removal involves exclusion of those words which do not add informative value in the sentences. After the stop words removal stemming aim in breaking the token into its base form. Index data structure will build on the obtain sequence of terms. These data structures are direct index, document index, lexicon and inverted index. These data structures contain different information about terms.

Index Data Structure	Contents
Lexicon	term
	term id
	document frequency
	term frequency
	byte and bit offsets in inverted index
Direct Index	term frequency
	term id gap (gamma code)
	fields (no. of field bits)
	block frequency
Document Index	document number
	document id
	document length
	byte and bit offsets in direct index
Inverted Index	term frequency
	document id gap (gamma code)
	fields (no. of field bits)
	block frequency (unary code)

Fig. 1 Index Structures and their contents

Direct index stores the term occur in the document in Fig. 2. It also stores the term frequency making the efficient query expansion. It also stores terms id, document id, fields in which term occur in the document and block id. Direct index is used while clustering the collection of documents. Document index store the information about document. It stores document number, document id, document length, length in byte and bit offset in direct index. Lexicon stores the information globally. It stores term, term id, document frequency, term frequency and

byte and bits offsets in inverted index. Inverted index stores the posting lists of the terms. It stores the term corresponding to the document in which it is store. It also stores the field of the terms. If index is based on positional information, then it also stores the block id which provides proximity search and phrasal search. After the document and terms are processed different indexing algorithm can be applied which can vary in terms of providing efficiency and memory storage.

Two-pass Indexing is divided into two parts. In first pass terms from the document are added into the temporary set or lexicon. These temporary different sets of lexicons are merged together to form a single lexicon. In the second pass with direct index and temporary lexicon inverted index is build. To convert direct index into inverted index several in memory iterations are performed. Inverted indexing requires large system storage as it stores both temporary lexicons and direct index. Whereas in single-pass indexing term posting are held in memory, and terms are merged to form lexicon inverted file. Single-pass index is faster than two-pass index as it does not construct direct index. In block index every small unit of document is considered as block generally of size 1. It stores the positional information of text in the document which facilitates phrasal search and proximity search.

Compression results in efficient information retrieval system. For laws regarding term this can be seen with the help of Heap's law and Zipf's law. Heap's law relates vocabulary size to the collection of documents. It states that the vocabulary size increases with the increase in the size of document. Zipf's law describes term distribution modelling in the collection of documents. It states that the frequency table ranking of term is inversely proportional to the term frequency of document.

Similarly, different techniques are used for dictionary compression. As heap's law states as collection size increase dictionary size also increases. Dictionary can be kept in memory in case of small collection size but for larger collection dictionary needs to be compressed. To preserve all the information which is needed in case of indexing lossless compression is used. The terms can be represented as a fixed length size string. These strings are

coupled with pointers. When strings are saved as terms it saves 60% of the storage space compared to fixed-width element arrays.

Other technique for compression is when document terms are saved as block storage. In this technique terms are grouped together in k sized block. Each block contains the pointer to the first term. For defining length terms addition byte can be store. As we increase the block size compression results gets better. Variable encoding byte encodes gap between posting using integral number of bytes. This is commonly of two type unary encoding and gamma encoding. Unary encoding contains n 1s that ends with 0. Gamma encoding uses length and offset to represent gap.

In a study by D. Minnie [8], different indexing algorithms and searching algorithm are intelligent search engine text document representation. It shows intelligent indexing technique by classifying the text in the document. Web crawler extract the information of the different web pages, index them and then use this information for web search engine. These contents are analysed and then it is indexed by the indexer. The web pages are selected with TFIDF (Term Frequency-Inverse Document Frequency). For efficient information retrieval data is collected, parsed, and stored in the index. Whenever user fires a query the search engine examines stored web pages with the help of an indexer and gives the list of best-matched results. Different indexing algorithms are used for the efficient searching process.

In a word count algorithm based on the occurrence of a particular word in a record the file is indexed. It also calculates the term frequency of a term present in the document. It split the sentences in to small units, removes stop words then calculates the occurrence of each word and save it to the database. The unique word algorithm calculates the inverse document frequency by separating the files on the basis of words found in them.

This algorithm parses the text present in the document, identify unique words ranks them, and then gives ranking to the document in the index table. In Comment Algorithm index the files based on the comments given by the writer which precisely describe the document. Here file is read and then comments are assigned to each file which is indexed along with the document. Bold Text Algorithm index the emphasized words present in the document. It selects the bold phrases from the document and index them. Italic Text Algorithm index italic terms from the document. Link Algorithm identifies the hypertext present in the document and index their links. Heading, Subheading Algorithm identifies heading and subheading of the document and index them.

For searching process exact query search algorithm m and interpreted query search algorithm is used. In exact query search algorithm user query is mapped with the contents of the document and matched results are displayed to the user. If user is not satisfied from the results then interpreted query search algorithm is used. It searches for the synonyms of the keyword along with the keyword.

METHODS

The word and phrase prediction tool examine all the documents and then search for the keywords providing a searching feature to the user as they enter the text [9]. Here we are focusing on developing a prediction tool which suggest the next words or phrase to the user, reducing the keystrokes as user need not to type the whole sentence. This reduces the fatigue in typing process by providing the better efficiency.

The System Architecture proposed consists of:

Searching blog

In Searching blog, documents of different format such as pdf, ppt, doc etc. will be uploaded and text will be extracted from these documents to implement indexing. To achieve efficient results inverted indexing is used. Text preprocessing will be performed to analyze the user entered query. Stop word will be removed and then stemming will be performed on the rest of the words. This will give relevant results on searching. Indexing will map the terms with the uploaded text document and gives the list of text documents which contain the fired query.

Output module

Output module will contain the retrieved results and returns the documents containing the keyword. It contains the line number in which the keyword is present along with whole sentence. The retrieved results will be displayed in different keyword indexing format such as KWIC, KWAC and KWOC.

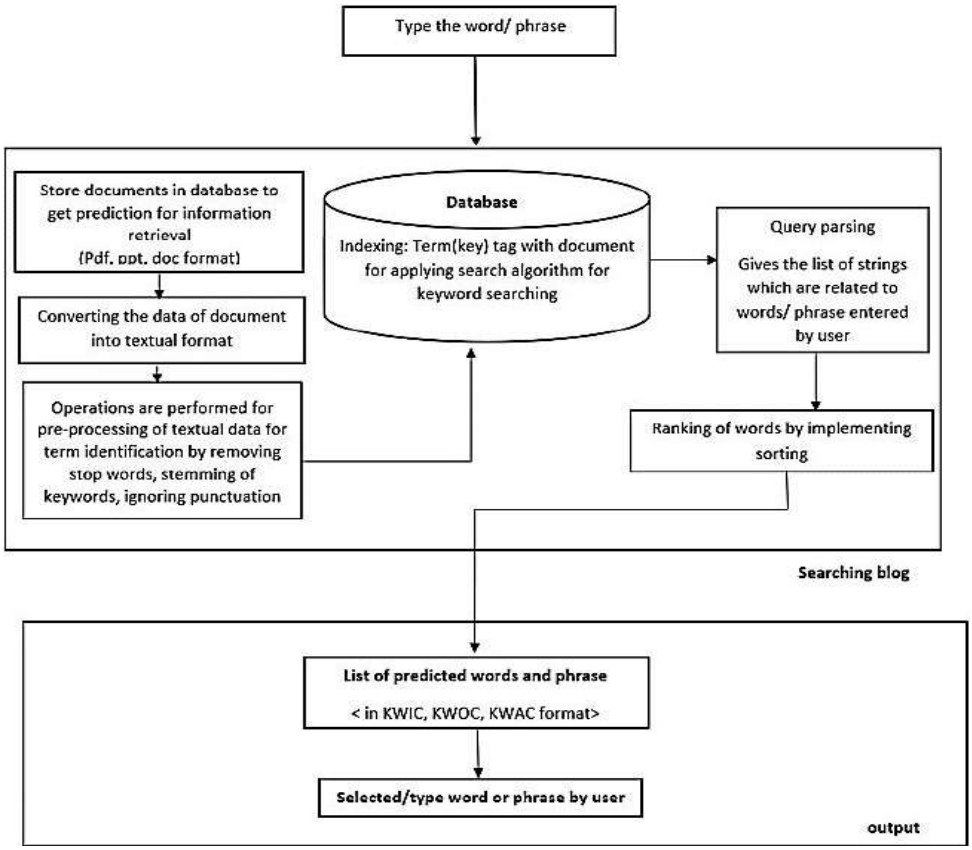


Fig. 2 System Architecture

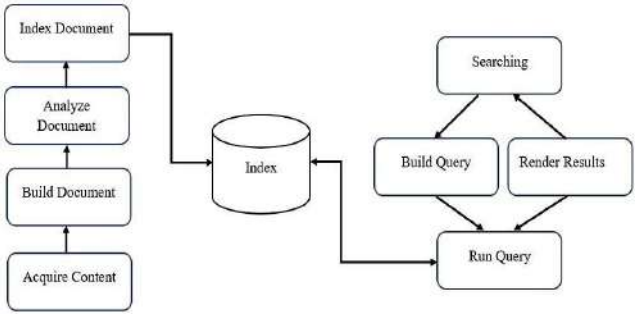


Fig. 3 Search System

SEARCHING BLOG

In the searching, the documents will be uploaded and then to extract relevant information from the uploaded file various text pre- processing operations will be performed. And then indexing of documents will be done for searching process to acquire efficient result. Main feature performed in searching blog are shown in Fig. 4:

Analyze Document: Text Document is a stream of text. Text is broken into small units called tokens to perform indexing.Indexing: In indexing process tokens are converted into index format for facilitating fast keywords look-ups.

Searching: It is the process of searching the index and retrieving the matching query.

Various steps that are performed in the searching blog are as follows:

1. Upload files:

Word prediction tool involves searching of keywords in file system. To upload files third party API used. This application programming interface (API) is a component that supports the file upload. API is a service provided by an application to exchange the data. Third party API reduces extra data entry to maintain multiple files. This provides multiple files uploading facility of various type such as pdf file, doc file, ppt etc.

Searching process involves following steps:

1. Identifying the words
2. Formatting the words
3. Turning the words
4. For obtain refine search use filters and limits.
5. Review the obtain result.

2. Text pre-processing

Text preprocessing is necessary to perform on the data which is entered by the user to transform words into numerical features to display desired results. Text preprocessing converts text into analyzable form. For term identification text preprocessing steps are:

A	It	These
About	Its	They
Again	Itself	This
All	Just	Those
Almost	km	Thus
Also	Made	To
Although	Mainly	Upon
Always	Make	Use
An	May	Used

Fig. 4. Some Common Stop-words

1. **Stop Words Removal:** Stop words are the words which does not add much meaning to a sentence that is why they can be removed from the sentences without sacrificing the essence of the query entered by the user such as ‘a’, ‘that’. These syntactic words carry little information, are not semantically meaningful in Fig. 5, and are effectively noise in the indexing process. Thus, ignored in the matching process. This process increases the performance of word and few tokens are left thus it increases the searching accuracy.

Phrase prediction tool as the size of data decreases, and hence searching time also decreases. After removal of stop words only

2. **Stemming:** It refers to normalizing the words into its base form or root form known as lemma. It plays an important role in information retrieval by recognizing, searching, and retrieving more forms of words, it returns many numbers of document containing the searched query. For example, the words affects, affection, affecting, affected will be converted into **affect**. Stemming is performed on the keywords which are left after

the removal of stop words entered by the user from the search query. Stemming will search for those keywords in documents containing words which starts with the stemmed words.

For efficient searching of stemmed keywords:

- **Keywords**

Example: Stemming cuts off the end part of the word using different algorithms as shown below:

The tool will search for all the documents which contains the words starting from "accelerat".

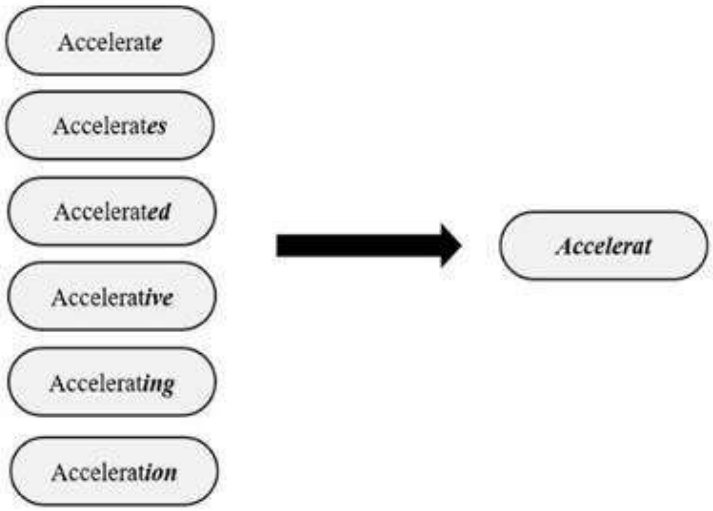


Fig. 5 Stemming

Word	Ends With	Remove Ending
Believes	Es	Believ
Consumption	Ion	Consumpt
Traceable	Able	Trace
Accountancy	Ancy	Account
Intoxicants	Ants	Intoxic
Problematic	Atic	Problem
Unlimitedly	Edly	Unlimit
Enlightened	Ened	Enlight
Rubbing	Ing	Rubb
Egyptians	Ians	Egypt
Gaseous	Eous	Gas
Backward	Ward	Back
Disgraceful	Ful	Disgrace

Fig. 6 General rules for stemming

3. Conversion of uploaded documents into textual format.

Keywords are the unique words or short phrases that captures the essence of the topic used to find information while searching. Keywords are used for indexing of document. These keywords are taken from title and abstract of document or text of the document. This process of keyword indexing also known as Natural Language Indexing and Full Text Indexing. Indexing system which generate index entries in unorganized structure are based on natural language indexing language of document. Text will be extracted from different kinds of uploaded having formats such as pdf, doc, ppt etc and then saved in text document format. Key terms indexing will be performed after the analysis of text present in the document(Fig. 9).

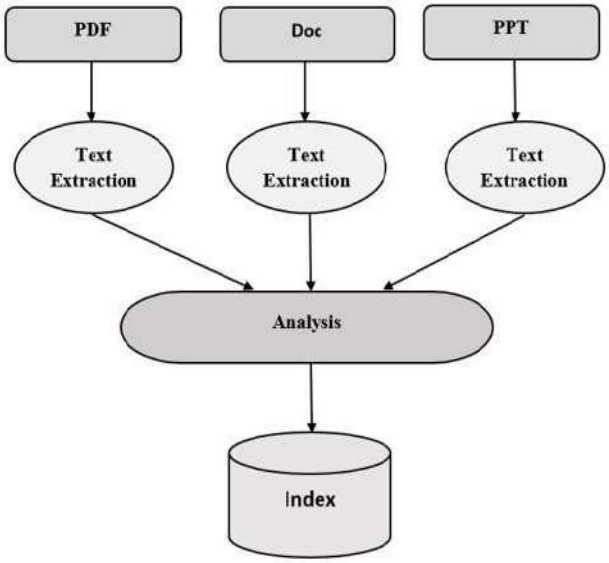


Fig. 7 Conversion into Textual Form

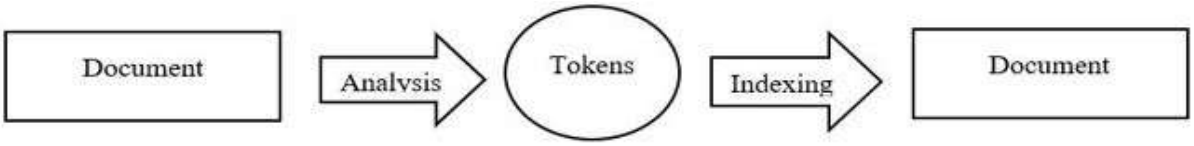


Fig. 8 Indexing flow

4. Indexing of textual data

After extracting the data from the documents, the resulting text will be analyzed. Analysis performs tokenization to transform text data into tokens. Tokenization is the process of operating strings into tokens which are in small structure and units. The tokens obtained from tokenization will be converted into terms, shown in Fig. 10, and used to build index. This process removes the punctuation marks from the text for efficient indexing. There can be some different cases where tokens can not be defined correctly. To overcome this problem tokenization is applies to both user query as well as indexed text documents.

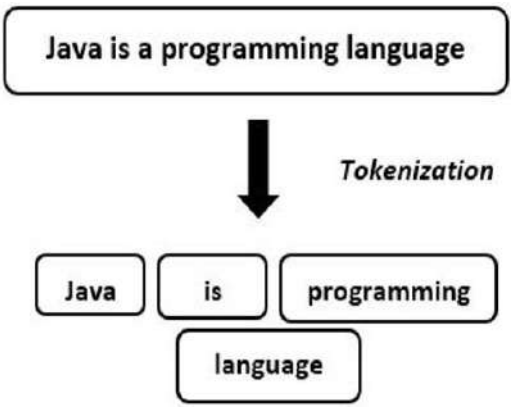
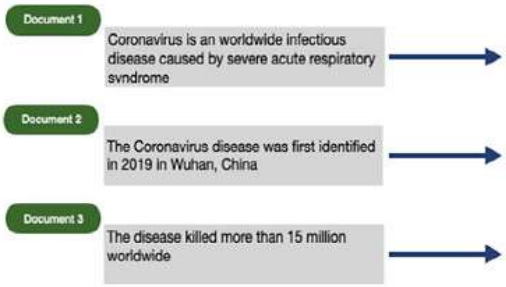


Fig. 10 Tokenization



ID	Term	Document
1	Coronavirus	1, 2
2	worldwide	1, 3
3	infectious	1
4	disease	1,2,3
5	severe	1
6	acute	1
7	respiratory	1
8	syndrome	1
9	first	2
10	2019	2
11	identified	2
12	Wuhan	2
13	China	2
14	killed	3
15	million	3

5. Creating indexes of text documents

When a query is processed indexing optimizes the performance by retrieving the result faster. Then the tool uses inverted indexing approach. It is an index data structure which stores the mapping from the content of documents, which means it maps the terms or keyword locations in the uploaded set of documents. It directs the user from a word to a document. Inverted indexing is of two types:

1. **Record-Level inverted indexing:** It includes the list of references to all the text documents for each keyword.
2. **Word- Level inverted indexing:** It includes the list of references to text documents for each keyword along with the location of the keyword in the text document.

The word and phrase prediction tool is based on inverted indexing and merge sort. Inverted indexing also referred as inverted file is a very simple data structure which stores mapping of content or word or numbers into file. It is very similar to the back of the book. The intent of using inverted index is to retrieve the searches result faster. To expedite the search process, we use inverted index. Here is an example explaining inverted indexing.

In fig. 11, if we are searching for “syndrome” inverted indexing would return document 1. And if we search for “disease” it would return document 1, 2 and 3. Here the words are in sorted order or in lexical order. These words are stores in tree format it can traverse and retrieve the index to locate the information much faster. This process is inverted indexing. The key aspect of inverted indexing is achieved at the cost of creating the index performance at the creation time. There are various ways of creating index and updating on index such as B-Tress and stagnated

or accumulated merge sort. Merge sort is very efficient and helps in avoiding continuous updates and give us better performance.

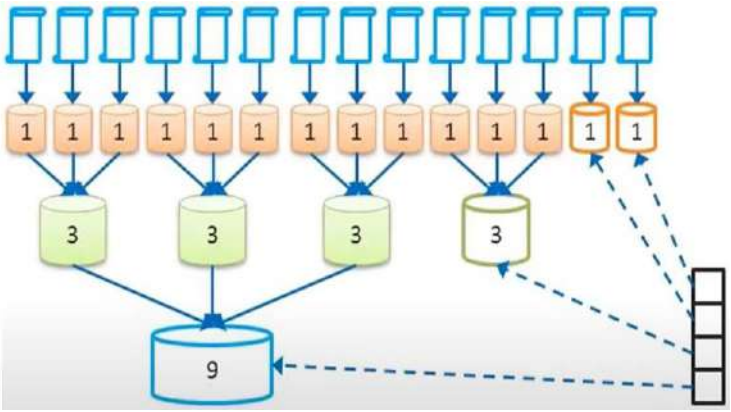
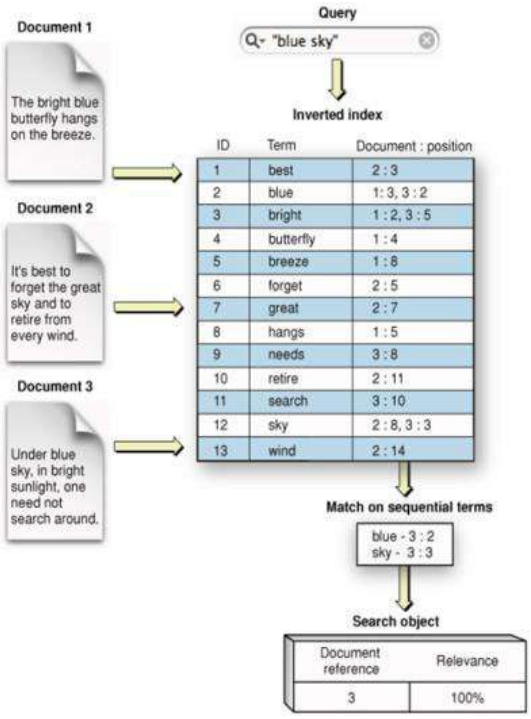


Fig. 11 Document Segmentation

For implementing inverted index let us say we have a list of documents in Fig.12. Here instead of storing all documents into one huge look up table and continuously keep on updating those lookup table. It creates segmented index look ups which is second level of categorising and then mapping those into a group of documents. In this way instead of creating huge set of segments in one part we create segments or partitions of the look up itself. So the look up itself is segmented and partitioned. Once we have segmented the look-ups then any newly created documents are created as new segment. If we update or add the documents in our system then a new segment is created and it will not be immediately added to the index rather it will be added as a local segment. Hence for adding a new set of documents instead of going and rewriting all the pages back again, inverted index writes the new documents into a blank new paper or new documents. So that page or paper is retained for some amount of time. Any new document will be written into the new page which we will call it as new segment. When we perform search during that period where the merge has not happened. The read has to locate those terms into the local page which is not yet merge to the global index. Once the threshold is crossed. It takes all the pages, apply merge sort and updates the index. This is the most efficient way of achieving the performance for information retrieval system.



In figure 13 we have three documents and each document has a stream of text. After the text is extracted from the input documents indexing is performed in textual format it stores the text document in index format which facilitates faster random access to the keywords stored in the document. The concept behind inverted indexing is similar to the end of a book, which helps in quickly locating the pages in the book that contains certain keywords. Indexing enables look-ups of words or phrase in a large file system and retrieves the set of documents that keywords entered by user. In the diagram it can be seen that only key terms such as butterfly, sky, wind etc has been extracted and indexed whereas common words like the, in, and, to etc has been omitted. The key terms which are indexed have a term id along with the offset of the location where the terms have occurred in the document. This enables the tool for fast full text searches in document retrieval.

6. Searching of query

The tool will search for the keywords in indexed file and returns the text documents which matches the query given by the user in organized manner based on the different data retrieval criteria. Thus, searching process will search for the keywords,

locate the document which contains those keywords in file system and gives the result which contains:

• the title of the document.

• the keywords found in the text document

• the sentence in which the keyword has been found in the document.

• location in which the searched word is found in the document

7. Ranking

The searching process will return the ordered set of documents in result containing keywords that matches the user query. Here sorting is implemented to rank the document according to the number of keywords found in each document. Those documents which contains the highest number of keywords will be ranked first.

OUTPUT

The words which are predicted by the tool can be displayed in three formats. These formats can be changed according to user preference.

These formats consist of three parts:

1. **Keyword**: The word which denotes the main subject of the document.
2. **Context:** The text which is surrounding to the keyword.
3. **Location Code or Identification:** Location of the document where it is available in the database represent by a numeric value.

The user will be having three buttons for selecting the desired format in which they want the result.

Example: The sentence "Java is a programming language". The given sentence will be shown in

A. KWIC (Keyword in Context) format as:

This method utilizes the computer capabilities and provides indexes solely derived from the title of the document. The selected keywords are printed in such a format that significs the selected keyword around them. It uses the stop-list to reduce the index content by removing the stop words from the title and highlight the keyword within the title along with the line number. The keywords selected are represented in such a way that it stands out from the rest of the contents. KWIC format increase the depth and range of indexing as it uses title and abstract for indexing process.

B. KWOC (Keyword out of Context) format as:

Here each keyword is taken out and print separately in the extreme left hind side of the unmodified sentence along with the complete sentence. Line number in which keyword is present is also mentioned. Sometimes, the selected key terms form the user query are printed as heading and the unmodified sentence printed in the next line as shown by the following example:

C. KWAC (Keyword Augmented in Context) format as:

It has been seen that the absolute dependency on the keyword sometimes fails in the retrieval of significant documents from the file system. KWAC format provides additional important keywords along with the selected keywords. The additional keywords are taken either from abstract or from the context of the document. KWAC format provides additional intellectual information and it is extended form of KWIC and KWOC format.

These are the different format in which result will be shown to the user and they can select the desired format according to the reference they seek

Evaluation Criteria

In the experimental evaluation of information retrieval systems, it can be seen that consistent development in indexing techniques and to evaluate indexing system precision and recall is used.

Recall: It is the index's capacity to retrieve the relevant document from the file system. It is the ratio of retrieve relevant document to the total number of relevant document available in the file system. It computes the extensiveness of the output documents and it is expressed as recall ratio.

Recall ratio = (k/N)*100

Where, k is the total number of relevant retrieved search documents.

N is total number of available documents against the particular search in the file system.

Precision: When query is fired the tool retrieves the irrelevant document to the user query along with the relevant document. Irrelevant document and leads to the time wastage as they affect the efficiency of the retrieved result of the tool as these results will be discarded by the user. Precision refers to the index capacity to obstruct the

document which are not relevant to the user query. It is the ratio of relevant retrieved document to the total number of documents retrieved by the tool. It computes the exactness of generated output and evaluate the functioning of indexing system.

Precision ratio = (k/M)*100

Where, k is the total number of relevant retrieved search documents.

M is total number of retrieved documents against the particular search in the file system which includes relevant and non-relevant documents.

CONCLUSION

As the resources are increasing day by day there is a need of efficient searching methods. Keyword Indexing and searching play a vital role in natural language indexing of text retrieval in file system with optimised speed and performance as it can use any term from the document to describe it. Without indexing we have to scan each file to obtain results which will consume a lot of time. Word and phrase prediction tool retrieve the search results by extracting the text from the file uploaded by the user. To obtain results faster, efficient inverted indexing technique is used for faster searching process. The result will contain the document which have the searched terms and displayed in KWIC, KWAC, and KWOC formats.

REFERENCES

Martin J, Word Prediction Technology: What it is and how it works. https://www.understood.org/en/school-learning/assistive-technology/assistive-technologies-basics/word-prediction-technology-what-it-is-and-how-it-works

Singh SP, Kumar A, Darbari H, Gupta S, Kanika (2016), Bilingual Keyword Indexing and Searching framework, International Conference of IEEE on Electrical, Electronics, and Optimization Techniques (ICEEOT)

Dudhabaware RS, Madankar M.S (2014), Review on natural language tasks for text document, International Conference of IEEE on Computational Intelligence and Computing Research

Mukherjee B (2017), Keyword indexing, eGyanKosh http://egyankosh.ac.in/handle/123456789/35769

Haidar S (2019), Derived Indexing, Library Studies and Information Technology http://egyankosh.ac.in/handle/123456789/35769

Premalatha.R, Srinivasan.S (2014), Text Processing in Information Retrieval System Using Vector Space Model, International Conference of IEEE on Information Communication and Embedded Systems (ICICES)

Kaur H, Gupta V (2016), Indexing process insight and evaluation, International Conference of IEEE on Inventive Computation Technologies (ICICT)

Minnie D, Srinivasan S (2011), Intelligent Search Engine Algorithms on Indexing and Searching ofText Documents using Text Representation. International Conference of IEEE on Recent Trends in Information System

Hatcher E, Cutting D (2004), Lucene in Action, Manning Publications Co., Greenwich, CT, United States of America, pp 56-100