RETRIEVAL OF SEMANTICALLY RELEVANT DOCUMENTS USING LATENT SEMANTIC ANALYSIS

CS6611 - CREATIVE AND INNOVATIVE PROJECT

Submitted by

NITISH K S (2020103550)

SATHYA NARAAYANAA S (2020103567)

VELMURUGAN J (2020103585)

in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

COLLEGE OF ENGINEERING, GUINDY

ANNA UNIVERSITY : CHENNAI 600 025

MAY 2023

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certificate that this project request titled Retrieval Of Semantically Relevant Documents Using Latent Semantic Analysis is the bonafide work of **NITISH K S (2020103550), SATHYA NARAAYANAA S (2020103567)** and **VELMURUGAN J (2020103585)** who carried out the project work under my supervision for the fulfillment of the requirements as part of the CS6611 – Creative and Innovative Project.

DR. S VALLI
PROFESSOR & HEAD
HEAD OF THE DEPARTMENT

Department of Computer Science
Science
and Engineering
Anna University
Chennai - 600025

MS. C. SUGANTHINI
TEACHING FELLOW
SUPERVISOR

Department of Computer

and Engineering
Anna University
Chennai - 600025

ABSTRACT

The Context-based Information Retrieval System (CBIRS) is designed to improve the accuracy and relevance of search results by leveraging advanced NLP techniques. The pre-processing step of CBIRS is critical in improving the quality of the article data before applying SVD on the TF-IDF matrix. The first technique used is stopword removal, which eliminates commonly occurring words in the English language, such as "the," "and," and "a," that do not carry much semantic value. This reduces the size of the data and helps to focus on more meaningful terms.

The next technique used in pre-processing is lemmatization, which reduces each word to its root form. This is particularly important for languages like English that have many morphological variations of words. For example, lemmatization would reduce words like "running," "runs," and "run" to the root form "run." This technique helps to reduce the dimensionality of the data and make it easier to extract relevant information.

The third technique used in preprocessing is lesser term frequency removal, which eliminates infrequently occurring terms that may not be relevant to the search context. This helps to further reduce the size of the data while preserving the important terms.

After pre-processing, CBIRS applies Singular Value Decomposition (SVD) on the TF-IDF matrix of the article data. SVD is a matrix factorization technique that reduces the dimensionality of the data while maintaining its semantic meaning. It decomposes the matrix into three parts: the left singular vectors, the singular values, and the right singular vectors. The singular values represent the most important

concepts in the data, and the left and right singular vectors represent the relationship between the terms and documents.

By using SVD, CBIRS is able to capture the underlying structure of the data and identify the most important concepts related to the search context. This results in more accurate and relevant search results for users, as it provides a more comprehensive understanding of the search context

ACKNOWLEDGEMENT

Foremost, we would like to express our sincere gratitude to our project guide, **Mrs. C. Suganthini**, Teaching Fellow, Department of Computer Science and Engineering, College of Engineering Guindy, Chennai for her constant source of inspiration. We thank her for the continuous support and guidance which was instrumental in taking the project to successful completion.

We are grateful to **Dr. S. Valli** , Professor and Head, Department of Computer Science and Engineering, College of Engineering Guindy, Chennai for her support and for providing necessary facilities to carry out for the project.

We would also like to thank our friends and family for their encouragement and continued support. We would also like to thank the Almighty for giving us the moral strength to accomplish our task.

NITISH K S SATHYA NARAAYANAA S VELMURUGAN J

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The growth of digital data in today's world has led to an increasing demand for effective information retrieval systems. While traditional keyword-based systems have limitations in accurately identifying relevant data, Context-based Information Retrieval Systems (CBIRS) have emerged as a solution.

CBIRS uses various techniques, such as web crawling and NLP pre-processing, to extract and cleanse relevant data from large datasets. Web crawling involves automatically traversing the internet to identify and extract e-news article data. The extracted data is then pre-processed using NLP techniques, such as stopword removal, lemmatization, and lesser term frequency removal, to cleanse the data and make it more easily searchable.

After pre-processing, CBIRS applies Singular Value Decomposition (SVD) on the Term Frequency-Inverse Document Frequency (TF-IDF) matrix of the article data. SVD is a matrix factorization technique that reduces the dimensionality of the data while maintaining its semantic meaning. By using SVD, CBIRS can capture the underlying structure of the data and identify the most important concepts related to the search context.

The use of SVD enables CBIRS to provide more accurate and relevant search results for users, as it provides a more comprehensive understanding of the search context. This can be particularly useful in complex contexts where traditional keyword-based systems may fail to identify relevant data.

Overall, CBIRS is an effective tool for filtering and extracting relevant information from large datasets, and its use of SVD enables it to capture the underlying structure of the data and provide more accurate search results. As the amount of digital data continues to grow at an exponential rate, CBIRS is likely to become even more important in the future.

1.1 PROBLEM STATEMENT

In today's digital age, finding relevant information quickly and efficiently has become a significant challenge due to the overwhelming amount of information available from various sources. Traditional information retrieval systems solely rely on keyword matching and often fail to deliver accurate results as they do not consider the context of the query or user. Context-based information retrieval takes into account various contextual factors such as the user's context and location, query/document context, and spatial-temporal context as well as the context of the information being searched, such as the language, domain, and time of creation.

1.2 OBJECTIVE

The objective of this system is to develop a retrieval system that can effectively retrieve semantically relevant documents using Latent Semantic Analysis (LSA). LSA is a mathematical technique that analyzes the relationships between terms and documents to capture the underlying semantic meaning. By employing LSA, the system aims to improve the accuracy and relevance of document retrieval by considering the context and meaning of the query rather than relying solely on keyword matching.

1.3 NEED FOR THE SYSTEM

The exponential growth of digital information has made it increasingly challenging for users to find relevant documents. Traditional keyword-based retrieval systems often struggle to capture the subtle nuances of language and context, resulting in inaccurate and overwhelming search results. There is a need for a retrieval system that can understand the semantic relationships between words and documents, enabling users to retrieve information that aligns with their intended meaning and context.

1.4 CHALLENGES OF THE SYSTEM:

Developing a retrieval system based on LSA poses several challenges. Firstly, creating a comprehensive and representative corpus of documents is crucial to ensure accurate semantic analysis. The system must be able to handle large volumes of data efficiently to provide timely retrieval results. Additionally, LSA relies on the assumption that the semantic structure of language can be captured in a low-dimensional space, which may not always hold true in complex linguistic contexts. Overcoming these challenges requires careful implementation of LSA algorithms, robust preprocessing techniques, and efficient indexing methods.

1.5 SCOPE OF THE PRODUCT

The product aims to provide a user-friendly interface for document retrieval, allowing users to input their queries and receive semantically relevant document suggestions. The system will incorporate preprocessing techniques to clean and normalize the textual data, followed by LSA-based analysis to capture the semantic relationships. The scope also includes developing an efficient indexing mechanism to

optimize retrieval speed. The product's applicability extends to various domains where accurate and contextually relevant document retrieval is essential, such as academia, research, digital libraries, and information-intensive industries.

CHAPTER 2

LITERATURE SURVEY

2.1 Techniques for improving Information Retrieval using NLP

In this paper [1] proposed an application research on Latent Semantic Analysis (LSA) for information retrieval. LSA is a statistical method used for extracting and representing the contextual-usage meaning of words in text documents. The advantage of using LSA in information retrieval is that it can effectively handle the problem of synonymy and polysemy. LSA can identify the latent semantic structure of a document and match it with the query to provide better results. However, the disadvantage of using LSA is that it requires a large amount of computational resources to process and is not suitable for real-time applications.

In this paper [2], evaluated stop word lists in text retrieval using Latent Semantic Indexing (LSI). LSI is a dimensionality reduction technique that uses singular value decomposition to identify the underlying relationships between words in a document. The advantage of using LSI in text retrieval is that it can process large volumes of documents and identify latent concepts. The study found that the removal of stop words improved the performance of LSI in information retrieval. However, the limitation of using stop word lists is that they may not be effective in capturing the context and meaning of the text.

In this paper [3], discussed similarity measure approaches applied in text document clustering for information retrieval. The study examined several similarity measures, including cosine similarity, Jaccard similarity, and Euclidean distance. The advantage of using similarity measures in

text document clustering is that it can group related documents together and improve the efficiency of information retrieval. However, the limitation of using similarity measures is that they may not capture the underlying semantic relationships between the documents.

2.2 Personalized Information Retrieval using Neural Networks

In this paper [4], proposed DSMN, a personalized information retrieval algorithm based on an improved Deep Structured Semantic Model (DSSM). DSMN uses a neural network to model the user's preferences and interests and provides personalized search results. The advantage of using DSMN is that it can improve the relevance of search results and provide a personalized search experience. However, the limitation of using DSMN is that it requires a large amount of data to train the neural network, and the performance may degrade if the user's preferences change.

2.3 Evaluation of Information Retrieval Models

In this paper [5], analyzed various information retrieval models, including Boolean model, vector space model, and probabilistic model. The advantage of using these models is that they provide a theoretical framework for information retrieval and can be used to develop efficient search algorithms. However, the limitation of these models is that they may not be able to capture the complex relationships between words in a document and may not provide accurate results in certain scenarios.

2.4 Summary

The reviewed papers focused on various information retrieval methods, including Latent Semantic Analysis, Latent Semantic Indexing,

similarity measures, and personalized information retrieval algorithms. The advantages identified included improved performance, efficient search algorithms, and personalized search experiences. However, the reviewed methods also have some limitations, such as high computational requirements, ineffective stop word lists, limitations in capturing underlying semantic relationships, and requirements for large amounts of data to train neural networks. These limitations may impact the accuracy and real-time performance of these methods in certain scenarios.

CHAPTER 3

SYSTEM DESIGN

The architecture diagram of the Context-based Information Retrieval System (CBIRS) is designed to efficiently and effectively extract relevant information from large datasets. It consists of four main modules that work together seamlessly to achieve this goal.

The first module is the dataset collection module, which involves web scraping to collect the data for analysis and storing it in a database. This module is responsible for retrieving e-news article data from the internet and storing it in a structured format for further processing.

The second module is the data preprocessing module, which includes several NLP techniques such as data cleaning, tokenization, stop word elimination, lemmatization, and Word Vectorization using tf-idf matrix algorithm. This module is responsible for cleansing and preparing the data for analysis by removing irrelevant words and extracting meaningful terms. The preprocessed data is then transformed into a TF-IDF matrix that represents the frequency of each term in each document.

The third module is the training model, which uses latent semantic analysis and does factor analysis to identify key concepts and relations between terms in the dataset. This module applies Singular Value Decomposition (SVD) to the TF-IDF matrix to extract the most important concepts and relations between the terms and documents.

Finally, the retrieval system module takes user queries, preprocesses them, and fetches relevant documents from the latent semantic analysis model using cosine similarity.

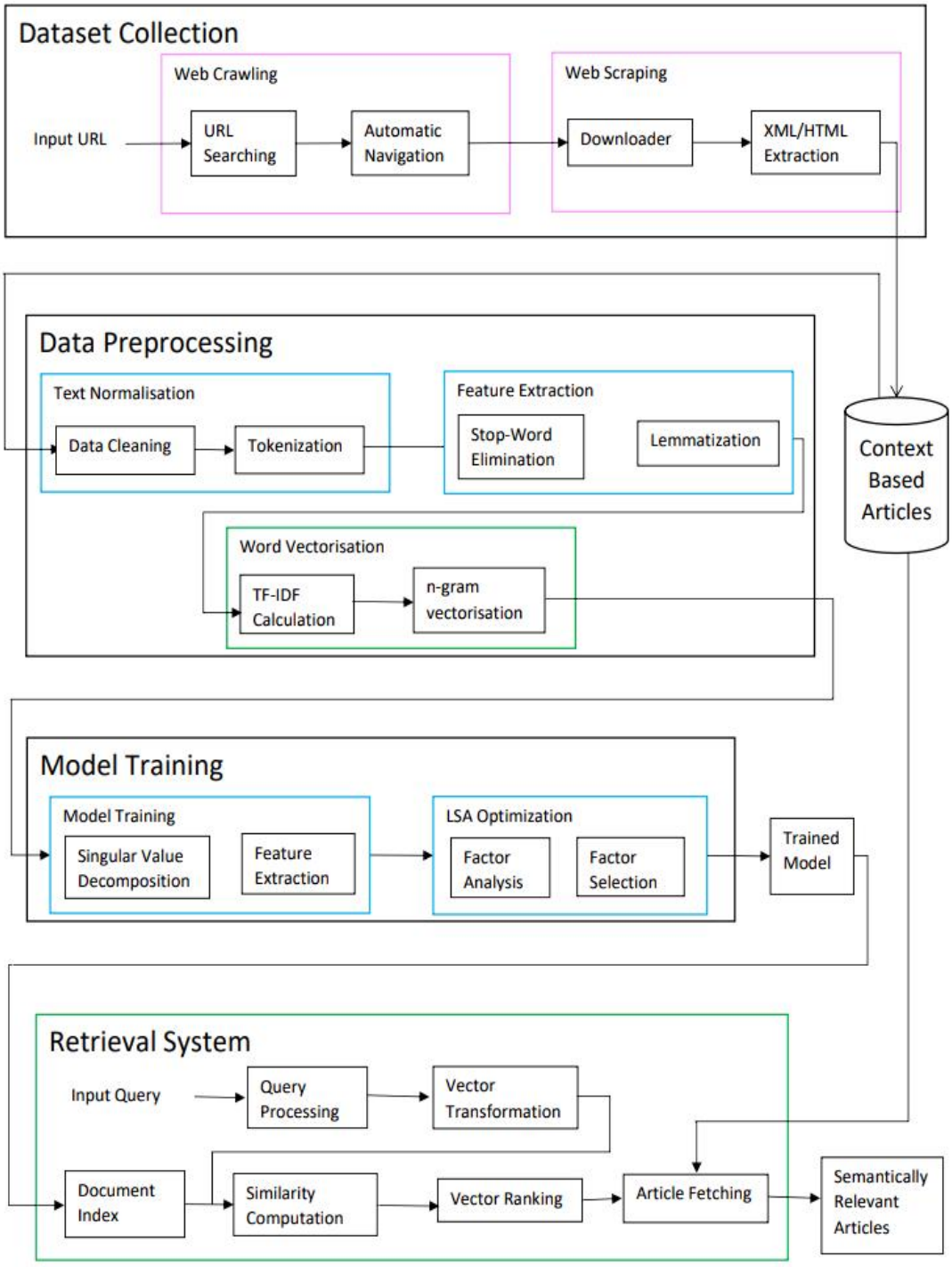
Dataset Collection
Web Crawling
Input URL
URL Searching
Automatic Navigation
Web Scraping
Downloader
XML/HTML Extraction
Data Preprocessing
Text Normalisation
Data Cleaning
Tokenization
Feature Extraction
Stop-Word Elimination
Lemmatization
Word Vectorisation
TF-IDF Calculation
n-gram vectorisation
Context Based Articles
Model Training
Model Training
Singular Value Decomposition
Feature Extraction
LSA Optimization
Factor Analysis
Factor Selection
Trained Model
Retrieval System
Input Query
Query Processing
Vector Transformation
Document Index
Similarity Computation
Vector Ranking
Article Fetching
Semantically Relevant Articles

Figure 3.1

3.1. SOFTWARE REQUIREMENTS SPECIFICATION

The scope of the project is to develop a system that can effectively search and retrieve context relevant articles based on user queries. The system will use Latent Semantic Indexing (LSI) to analyse and index the articles, enabling it to identify and retrieve articles that are semantically similar to the user's query, even if the articles do not contain the exact query terms. The project aims to improve the search experience by providing a more accurate and efficient way to find articles of interest.

3.1.1. FUNCTIONAL REQUIREMENTS:

3.1.1.1. Hardware Requirements

Processor: The application should require a processor with a minimum of 2 GHz clock speed and 4 cores.

Memory: The application should require a minimum of 2GB of RAM.

Storage: Sufficient storage space to store the articles and the indexing data.

3.1.1.2. Software Requirements:

Python: Version 3.7 or more

Nltk: Version 3.0 or more

Scrapy: Version 2.5 or more

Scikit-learn: Version 1.0 or more

3.1.1.3. Software Interfaces:

An API to allow external systems to access the system's functionality.

An user interface for users to enter their search queries and view the results.

A database to store the articles and indexing data.

Application is intended to work on all platforms including windows, macOs, and Linux.

The application should be able to use popular natural language libraries like nltk

3.1.2. NON-FUNCTIONAL REQUIREMENTS:

Performance: The system should be designed to handle large volumes of e-news articles and user requests, with minimal latency and response time.

Scalability: The system should be scalable, allowing it to handle increasing volumes of e-news articles and user requests without compromising performance.

Reliability: The system should be reliable, ensuring that it is available for users to access when required, with minimal downtime or outages.

Security: The system should be designed with security in mind, with measures in place to prevent unauthorised access, protect user data, and ensure data privacy.

Usability: The system should be easy to use and navigate, with a user-friendly interface that enables users to search and retrieve e-news articles quickly and efficiently.

Accessibility: The system should be designed to be accessible to users with disabilities, with features such as keyboard navigation, screen reader support, and colour contrast options.

Maintainability: The system should be easy to maintain and update, with well-documented code and a modular architecture that enables developers to make changes without disrupting the system's functionality.

CHAPTER 4

MODULE DESCRIPTION

4.1. Dataset Collection

This involves web scraping to collect data for analysis. Web scraping is the process of extracting information from websites and converting it into a structured format such as json that can be used for analysis. The collected data is then stored in a database, which can be accessed and analysed by the system. This process allows for the efficient and automated collection of large amounts of data, which can be used to train the latent semantic analysis model and improve the accuracy of the information retrieval system.

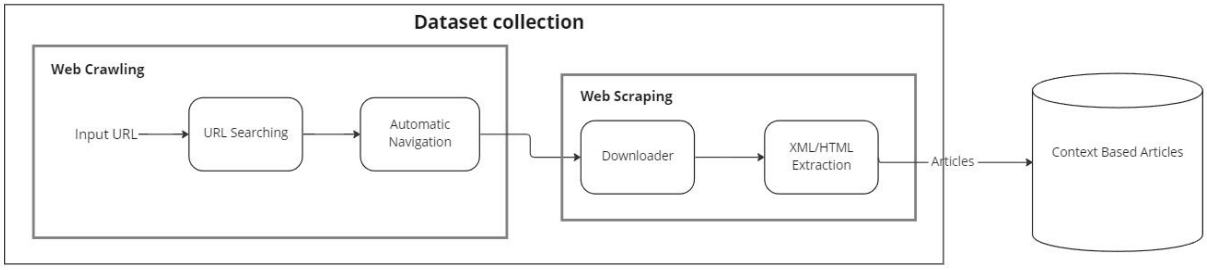

Figure 4.1

Input: Website URLs used to navigate and retrieve web documents.

Output: The retrieved documents, context-based stored in a structured format(JSON).

```
class myspider extends scrapy.Spider
    name <- spider_name
    start_urls <- Array(urls)
    parse (response):
```

```
for hrefs in response.css('a::attr(href)'):
    yield response.follow(href, parse)
for quote in response.css('div.quote'):
    yield {
  'text': quote.css('span.text::text').get(),
  'author': quote.css('span small::text').get(),
  'tags': quote.css('div.tags a.tag::text').getall(),
  }
```

4.2. Data Preprocessing

Data preprocessing is a crucial step in natural language processing. It involves several techniques such as data cleaning, tokenization, stop word elimination, lemmatization, and word vectorization using the tf-idf matrix algorithm. Data cleaning is used to remove irrelevant information and correct spelling mistakes. Tokenization is used to split text into individual tokens. Stop word elimination removes commonly used words such as "the" or "a". Lemmatization is used to reduce words to their base form. Word vectorization is the process of converting text into a numerical representation using the tf-idf matrix algorithm, which calculates the importance of each word in a document.

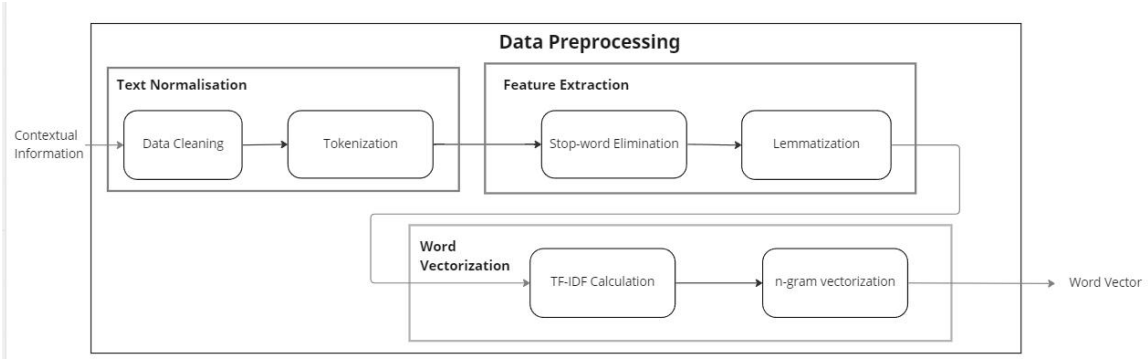

Figure 4.2

Input: Contextual information retrieved from web documents

Output: Word vector containing vector representation of key features

Text Normalisation and Feature Extraction

```
function preprocess(document):
   document <- regex_replace(document, r'[^a-zA-Z\s]', '')
   tokens <- word_tokenize(document)
   for token in tokens:
        if token not in stopwords('english')
         new_tokens += token
  tokens <- new_tokens
   lemmatizer <- WordNetLemmatizer()
   for token in tokens
        new_tokens.append(lemmatizer.lematize(token))
  tokens <- new_tokens
  return tokens
```

Word Vectorisation

```
term_frequencies <- []
for each article in articles:
   term_frequencies.append(calculate_term_frequency(article))
   terms.append(article.split())
inverse_document_frequencies <- {}
for each term in terms:
   inverse_document_frequencies['term'] <-
calculate_inverse_document_frequency(term)
tfidf_weights <- {}
for article in articles:
   for term in terms:
```

```
            tfidf_weights[(document, term)] <-
calculate_tfidf_weight(document, term)
            tfidf_matrix <- convert_to_matrix(tfidf_weights)
```

4.3. Model Training

Latent Semantic Analysis (LSA) is a mathematical technique used to extract and analyze the underlying relationships between words and documents. In this module, the preprocessed data is used to train the LSA model, which involves the calculation of the singular value decomposition of the term-document matrix. Factor analysis techniques such as factor rotation and factor extraction are also applied to the model to improve its accuracy and efficiency in retrieving relevant documents based on user queries. The resulting model is used in the retrieval system to rank and rate the relevance of retrieved documents.

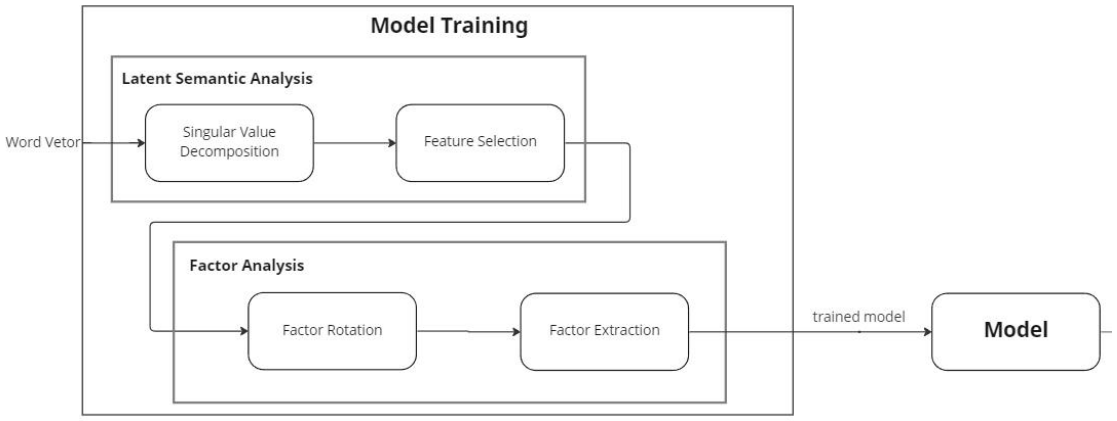

Figure 4.3

Input: Word vectors obtained from TF-IDF Matrix

Output: Trained model that can retrieve relevant documents based on a given query.

Latent Semantic Analysis

```
function svd(A):
    ATA <- np.dot(A.T, A)
    eigenvalues, eigenvectors <- np.linalg.eig(ATA)
    singular_values <- np.sqrt(eigenvalues)
    U <- eigenvectors
    V <- np.dot(A, U) / singular_values.reshape(-1, 1)
    return U, singular_values, V.T
```

Factor Analysis

```
function select_n_components(var_ratio, goal_var) :
    total_variance <- 0.0
    n_components <- 0
    for explained_variance in var_ratio:
        total_variance +<- explained_variance
        n_components +<- 1
        if total_variance ><- goal_var:
            break
    return n_components
```

4.4. Retrieval System

The retrieval system involves processing user queries and identifying relevant articles from a large database of information. This process typically involves several steps, including query processing, article retrieval, and ranking. First, the user query is processed to identify the key terms and concepts that are relevant to the search. Next, the system uses a trained model from latent semantic analysis (LSA) to identify articles that are semantically related to the user query. Finally, the ranked articles are presented to the user as the final output. This might

involve displaying the articles in a list or table, along with a summary or snippet of the content, to help the user quickly identify the most relevant results.

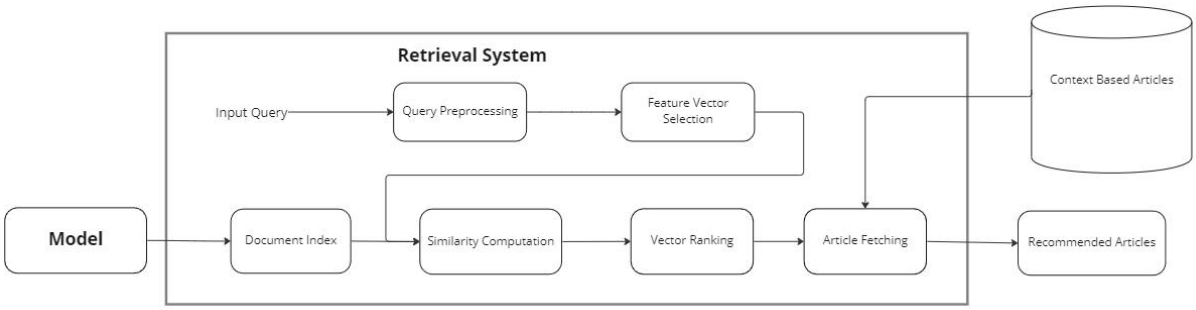

Figure 4.4

Input: User query

Output: Recommended articles

```
function retrieve_documents(query, documents):
    term_document_matrix <- create_term_document_matrix(documents)
    U, S, V <- svd(term_document_matrix)
    k <- 100
    U_prime <- U[:, :k]
    V_prime <- V[:k, :]
    term_document_matrix_reduced <- np.dot(U_prime, V_prime)
    query_vector <- create_query_vector(query)
    similarity_scores <- []
    for document_vector in term_document_matrix_reduced:
        similarity_scores.append(cosine_similarity(query_vector,
document_vector))
    retrieval_results <- []
    for i in np.argsort(similarity_scores)[::-1]:
        retrieval_results.append(documents[i])
    return retrieval_results
```

CHAPTER 5

RESULTS & IMPLEMENTATION

5.1. Dataset Description

The aim of our project is to retrieve relevant news articles based on the context of the user query. To achieve that we extract current news articles from various sources. The dataset consists of json files each consisting of articles from a specific source of different genre. Each file follows the below format.

```
[
    {
        "Heading": string
        "author": string,
        "publish_date": datetime,
        "overview": string,
        "link": url,
        "content": Array(string)
          },
    ...
]
```

5.2. Dataset Collection

In this module, the articles are extracted from the web and saved to their respective json files following the above format.

Code

Write spider to extract articles recursively

```
from pathlib import Path
import scrapy
class IndianExpSpider(scrapy.Spider):
name = "indian-express"
    start_urls = [
            'https://indianexpress.com/section/india/',
            'https://indianexpress.com/section/entertainment/',
                'https://indianexpress.com/section/political-
        pulse/',
            'https://indianexpress.com/section/technology/',
            'https://indianexpress.com/section/sports/'
          ]
```

URL Searching:

```
def parse(self, response):
    for newsItem in response.css('div.articles'):
      href = newsItem.css('h2 a::attr(href)').get()
      yield from self.downloader(response, newsItem, href)
```

Automatic Navigation:

```
def navigator(self, response):
      nextPage = response.css(
        'ul.page-numbers a.next::attr(href)').get()
      if nextPage is not None:
        yield response.follow(nextPage, callback=self.parse)
```

Downloader:

```
def downloader(self, response, newsItem, href):
        contentPage = response.follow(
            href, callback=self.extractor, cb_kwargs=dict())
        contentPage.cb_kwargs['heading'] = newsItem.css(
            'h2.title a::text').get()
        contentPage.cb_kwargs['author'] = ""
        contentPage.cb_kwargs['publish_date'] = newsItem.css(
            'div.date::text').get()
        contentPage.cb_kwargs['overview'] = newsItem.css(
            'p::text').get()
        contentPage.cb_kwargs['link'] = newsItem.css(
            'h2.title a::attr(href)').get()
        yield contentPage
        yield from self.navigator(response)
```

XML/HTML Extraction

```
def extractor(self, response, heading, author, publish_date, overview, link):
        yield {
            'heading': heading,
            'author': author,
            'publish_date': publish_date,
            'overview': overview,
            'link': link,
            'content': response.css('div.full-details p::text').getall()
```

Input format: (Specified in start_urls)

'https://indianexpress.com/section/india/',

'https://indianexpress.com/section/entertainment/',

'https://indianexpress.com/section/political-pulse/ ',

'https://indianexpress.com/section/technology/',

'https://indianexpress.com/section/sports/'

Start crawling using command

$ scrapy crawl spider-name -O articles/output.json

Output

Articles Extracted

{

"heading": "Mallikarjun Kharge slams ED searches",

"author": "",

"publish_date": "March 11, 2e23 e8:37 1ST",

"overview": "Where were the agencies of the Modi at Yadav family government when premises, fugitives says Modi govt trying to kill democracy ran away from the country with crores, Kharge asked",

"link": "https: // indianexpress.com/article/india/mallikarjun-kharge-slams-ed-searches-at-yadav-family-premises-modi-govt-kill-democracy",

"content": [

"Congress President Mallikarjun Kharge on Friday accused the

Narendra Modi government of making sinister attempts to kill democracy",

"on the premises of former Bihar Chief Minister Lalu Prasad Yadav's family. ",

"The Enforcement Directorate on Friday conducted searches in multiple cities of Bihar and other locations",

"The ED in its raids seized Rs 53 lakh, USD I, 90, about 54B grams of gold and bullion and 1.5 kg of gold jewellery, sources in the",

"where Lalu Prasad's son, Bihar Deputy Chief Minister, Tejashwi Yadav was present, they said.",

"Reacting to the searches, Kharge in a tweet in Hindi said",

"For the last 14 hours, Modi ji has kept ED at the house of Deputy Chief",

"Where were the agencies of the Modi government when fugitives ran away from the country with crores, he asked."

]

}

Log Information

2023-03-11 11: 03: 25 [scrapy.statscollectors] INFO: Dumping Scrapy stats:

{

' downloader/exception_count' : 2

'downloader/request_method_count/GET' : 1491,

'downloader/response_bytes' : 169494425,

'downloader/response _ count' : 1489,

'downloader/response status count/200': 1489,

'dupefilter/filtered' : 1479

'elapsed_time_seconds' : 150.264019,

'feedexport/success_count/Fi1eFeedStorage' : 1,

'finish reason': 'finished' ,

'finish_time': datetime.datetime(2023, 3, 11, 5, 33, 25, 24099),

'httpcompression/response_bytes' : 985709082,

'httpcompression/response count' : 1489,

'item_scraped_count' : 1438,

'log_count/DEBUG': 2933,

'log_count/1NFO': 13,

'request_depth_max': 50,

'response received_count' : 1489,

'retry/count': 2,

'retry/reason count/twisted.internet.error.DNSLookupError': 2,

'robotstxt/request_count' : 1,

'robotstxt/response_count' : 1,

'robotstxt/response : 1,

'scheduler/dequeued' : 1488,

'scheduler/dequeued/memory' : 1488,

'scheduler/enqueued' : 1488,

'scheduler/enqueued/memory' : 1488,

'start_time': datetime.datetime(2023, 3, 11, 5, 30, 54, 70080)

}

2023-03-11 11:03:25 [scrapy.core.engine] INFO: Spider closed (finished)

Total Fetched

python3 checkcount.py

The hindustan.json contains 1438 articles.

The indian exp.json contains 3054 articles.

The ndtv.json contains 190 articles.

The vox.json contains 264 articles.

The wion.json contains 14 articles.

5.2. Dataset Preprocessing

In this module, we cleanse and prepare the data for applying latent semantic analysis treatment to it.

Code

Data cleansing

```
def remove_punctuations(text):
    """Removes punctuation from text"""
    text = text.strip()
    text = text.translate(str.maketrans('', '', string.punctuation))
    text = re.sub(r'[^\w\s]', '', text)
    return text


def decontact(phrase):
    """Removes apostrophe word and numbers"""
    phrase = re.sub(r'\b\d+\b', '', phrase)
    phrase = re.sub(r"won't", "will not", phrase)
    phrase = re.sub(r"can\'t", "can not", phrase)
    phrase = re.sub(r"n\'t", " not", phrase)
    phrase = re.sub(r"\'re", " are", phrase)
    phrase = re.sub(r"\'s", " is", phrase)
    phrase = re.sub(r"\'d", " would", phrase)
    phrase = re.sub(r"\'ll", " will", phrase)
    phrase = re.sub(r"\'t", " not", phrase)
    phrase = re.sub(r"\'ve", " have", phrase)
    phrase = re.sub(r"\'m", " am", phrase)
    return phrase
```

Tokenisation

```
def generate_tokens(text):
    """Generates tokens using tokenizer"""
    text = text.lower()
    tokens = tokenizer.tokenize(text)
    texts = [word for word in tokens if word not in stopword_list]
    # texts = ' '.join(texts)
    return texts


def get_wordnet_pos(word):
    """Map POS tag to first character lemmatize() accepts"""
    tag = nltk.pos_tag([word])[0][1][0].upper()
    tag_dict = {"J": wordnet.ADJ,
                "N": wordnet.NOUN,
                "V": wordnet.VERB,
                "R": wordnet.ADV}
   return tag_dict.get(tag, wordnet.NOUN)


def capture_lemmatization(tokens):
    """Captures lemmatization and translates word accordingly"""
    tokens = [lemmatizer.lemmatize(
        token, get_wordnet_pos(token)) for token in tokens]
    return tokens
```

TF-IDF Vectorisation

```
def generate_unigram():
    vectorizer = TfidfVectorizer(ngram_range=(1,1))
    vectorizer.fit(article_vec)
    tfidf_matrix = vectorizer.transform(article_vec)

def generate_bigram():
    vectorizer = TfidfVectorizer(ngram_range=(2,2))
    vectorizer.fit(article_vec)
    tfidf_matrix = vectorizer.transform(article_vec)
```

Input:

Heading in an article:

Pro-Khalistan actors using gangs, nexus firmed up in jails in India and abroad, NIA says in chargesheet

First time in four years, NREGS jobs in Jan-Feb below pre-Covid level

Rahul Gandhi to Lok Sabha Speaker: 'Scurrilous, defamatory… let me respond to BJP'

Assam Class 10 board exam paper leak govt's failure: CM Himanta Biswa Sarma

Minors assaulted, their hair chopped off at Chhattisgarh residential school, eight students booked one sent to jail

Govt nod to acquire defence hardware worth Rs 70,500 crore

'Rigid stance': RS Chairman Dhankhar's meeting with ministers, Oppn leaders fails to break House impasse

PFI recruits underwent 3-stage training programme: NIA chargesheet
J&K leaders meet EC, ask for Assembly elections to be announced
Amritpal Singh still on the run: Centre asks BSF, SSB to be alert at border posts

Output

After removal of punctuations, apostrophes and numbers:

ProKhalistan actors using gangs nexus firmed up in jails in India and abroa
NIA says in chargesheet
First time in four years NREGS jobs in JanFeb below preCovid level
Rahul Gandhi to Lok Sabha Speaker Scurrilous defamatory let me respor to BJP
Assam Class board exam paper leak govt is failure CM Himanta Bisv Sarma
Minors assaulted their hair chopped off at Chhattisgarh residential scho eight students booked one sent to jail
Govt nod to acquire defence hardware worth Rs crore
Rigid stance RS Chairman Dhankhar is meeting with ministers Oppn leade fails to break House impasse
PFI recruits underwent stage training programme NIA chargesheet
JK leaders meet EC ask for Assembly elections to be announced
Amritpal Singh still on the run Centre asks BSF SSB to be alert at bord posts

Generation of tokens:

['prokhalistan', 'actors', 'using', 'gangs', 'nexus', 'firmed', 'up', 'in', 'jails', 'i 'india', 'and', 'abroad', 'nia', 'says', 'in', 'chargesheet']

['first', 'time', 'in', 'four', 'years', 'nregs', 'jobs', 'in', 'janfeb', 'below', 'precovi
'level']
['rahul', 'gandhi', 'to', 'lok', 'sabha', 'speaker', 'scurrilous', 'defamatory', 'le
'me', 'respond', 'to', 'bjp']
['assam', 'class', 'board', 'exam', 'paper', 'leak', 'govt', 'is', 'failure', 'cn
'himanta', 'biswa', 'sarma']
['minors', 'assaulted', 'their', 'hair', 'chopped', 'off', 'at', 'chhattisgarl
'residential', 'school', 'eight', 'students', 'booked', 'one', 'sent', 'to', 'jail']
['govt', 'nod', 'to', 'acquire', 'defence', 'hardware', 'worth', 'rs', 'crore']
['rigid', 'stance', 'rs', 'chairman', 'dhankhar', 'is', 'meeting', 'with', 'minister
'oppn', 'leaders', 'fails', 'to', 'break', 'house', 'impasse']
['pfi', 'recruits', 'underwent', 'stage', 'training', 'programme', 'ni
'chargesheet']
['jk', 'leaders', 'meet', 'ec', 'ask', 'for', 'assembly', 'elections', 'to', 'b
'announced']
['amritpal', 'singh', 'still', 'on', 'the', 'run', 'centre', 'asks', 'bsf', 'ssb', 'to', 'b
'alert', 'at', 'border', 'posts']

After stopword removal and lemmatization:

*['prokhalistan', 'actor', 'use', 'gang', 'nexus', 'jail', 'india', 'abroad', 'ni
'say', 'chargesheet']*
['first', 'time', 'four', 'year', 'nregs', 'job', 'level']
['rahul', 'gandhi', 'lok', 'sabha', 'speaker', 'let', 'respond', 'bjp']
*['assam', 'class', 'board', 'exam', 'paper', 'leak', 'govt', 'failure', 'cn
'himanta', 'biswa', 'sarma']*
*['minor', 'assault', 'hair', 'chopped', 'chhattisgarh', 'residential', 'schoo
'eight', 'student', 'book', 'one', 'sent', 'jail']*

['govt', 'nod', 'acquire', 'defence', 'worth', 'r', 'crore']

['rigid', 'stance', 'r', 'chairman', 'dhankhar', 'meeting', 'minister', 'oppr

'leader', 'fails', 'break', 'house', 'impasse']

['pfi', 'recruit', 'stage', 'training', 'programme', 'nia', 'chargesheet']

['jk', 'leader', 'meet', 'ec', 'ask', 'assembly', 'election', 'announce']

['amritpal', 'singh', 'still', 'run', 'centre', 'asks', 'bsf', 'ssb', 'alert', 'borde

'post']

Figure 5.1

Unigram Vectorization

{'prokhalistan': 5790, 'actor': 115, 'use': 7863, 'gang': 2919, 'nexus': 503

'jail': 3765, 'india': 3537, 'abroad': 53, 'nia': 5041, 'say': 656

'chargesheet': 1308, 'time': 7535, 'year': 8222, 'nregs': 5114, 'job': 384

'level': 4243, 'rahul': 5945, 'gandhi': 2915, 'lok': 4325, 'sabha': 643

'speaker': 7041, 'let': 4239, 'respond': 6235, 'bjp': 895, 'assam': 535, 'class

1424, 'board': 946, 'exam': 2474, 'paper': 5341, 'leak': 4195, 'govt': 306

'failure': 2586, 'cm': 1460, 'himanta': 3326, 'biswa': 886, 'sarma': 653 'minor': 4711, 'assault': 538, 'hair': 3177, 'chopped': 1376, 'chhattisgarh 1346, 'residential': 6223, 'school': 6589, 'student': 7203, 'book': 966, 'sen 6689… }

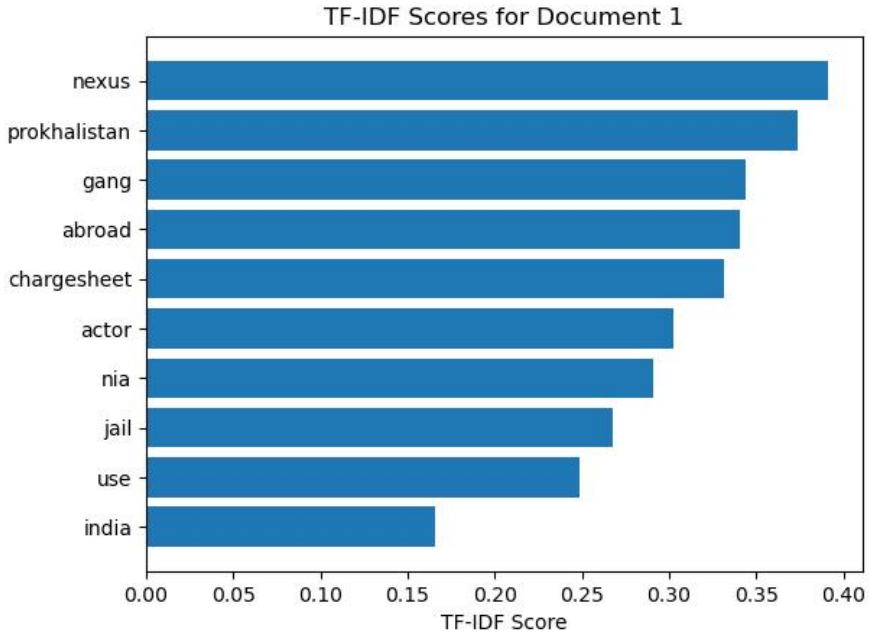

Figure 5.2

Bigram Vectorization

{'prokhalistan actor': 47030, 'actor use': 752, 'use gang': 63660, 'gan nexus': 23395, 'nexus jail': 40791, 'jail india': 30314, 'india abroad': 2855 'abroad nia': 277, 'nia say': 40843, 'say chargesheet': 52742, 'time year 61266, 'year nregs': 67370, 'nregs job': 41190, 'job level': 30762, 'rah gandhi': 48274, 'gandhi lok': 23312, 'lok sabha': 34929, 'sabha speaker 52159, 'speaker let': 57052, 'let respond': 34062, 'respond bjp': 5055 'assam class': 3751, 'class board': 11425, 'board exam': 7275, 'exam paper 19723, 'paper leak': 42965, 'leak govt': 33772, 'govt failure': 24392, 'failu cm': 20562, ...}

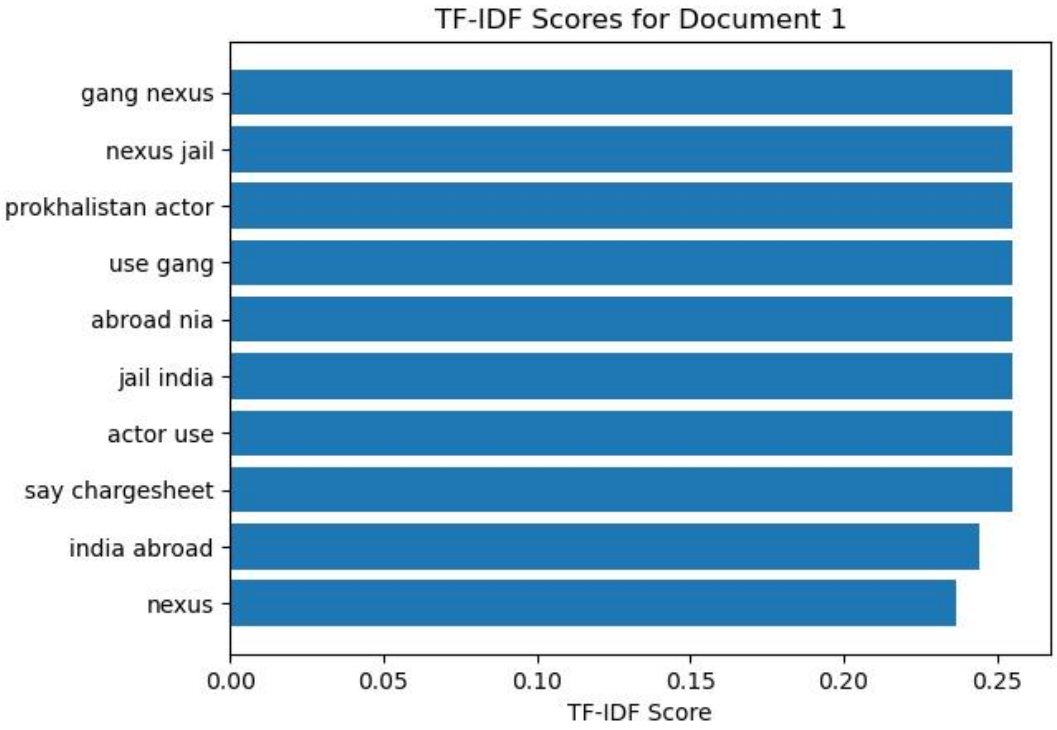

Figure 5.3

5.3 Model Training

In this module we train the LSA model and increase its optimacy.

Code

Singular Value Decomposition

```
iter = 10

svd = TruncatedSVD(n_components=7500,
algorithm='randomized', n_iter=iter, random_state=42)

for i in tqdm(range(iter)):

  lsa = svd.fit_transform(tfidf_matrix)

Sigma = svd.singular_values_

V_T = svd.components_.T
```

Factor Analysis

```
def select_n_components(var_ratio, goal_var: float) -> int:
    total_variance = 0.0
    n_components = 0
    for explained_variance in var_ratio:
        total_variance += explained_variance
        n_components += 1
        if total_variance >= goal_var:
            break
    return n_components
print("No of components for 0.75 variance: ",
select_n_components(svd_modeling.explained_variance_ratio_, 0.75))
```

Output:

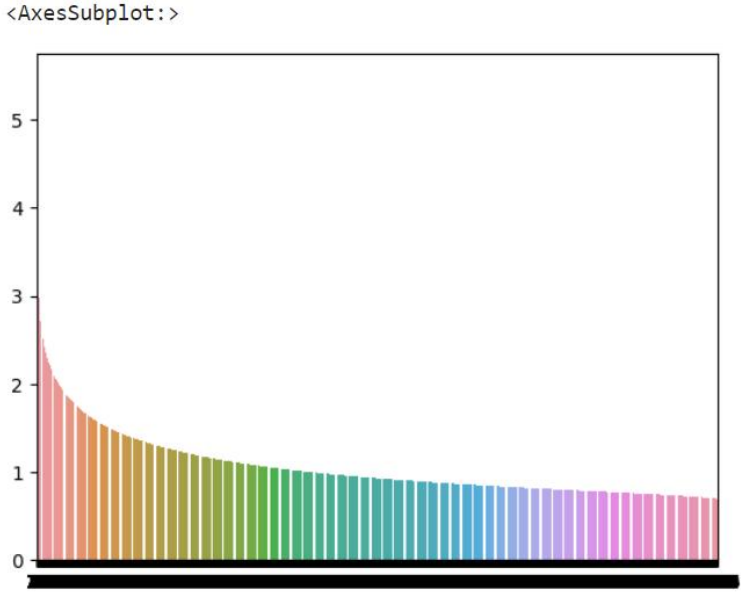

Figure 5.4

Topics

Topic 0: ['bjp', 'rahul', 'say', 'gandhi', 'india']

Topic 1: ['india', 'au', 'ind', 'ind au', 'world']

Topic 2: ['rahul', 'gandhi', 'rahul gandhi', 'au', 'ind']

Topic 3: ['quote', 'image', 'wish', 'status', 'photo']

Topic 4: ['world', 'cup', 'world cup', 'woman', 't20']

Topic 5: ['modi', 'pm', 'pm modi', 'bjp', 'narendra']

Topic 6: ['case', 'pm', 'modi', 'covid', 'pm modi']

Topic 7: ['covid', 'case', 'india', 'new', 'covid case']

Topic 8: ['delhi', 'say', 'govt', 'indian', 'watch']

Topic 9: ['delhi', 'hc', 'govt', 'world', 'woman']

Topic 10: ['delhi', 'gujarat', 'watch', 'wpl', 'day']

svd_model data

svd_modeling.explained_variance_ratio_:

array([1.00292821e-03, 1.78370618e-03, 1.49121145e-03, ...,
3.91293442e-05, 3.90956381e-05, 3.90867156e-05])

svd_modeling.explained_variance_ratio_.sum() : 0.7680815829865291

No of components for 0.75 variance: 7066

5.4 Retrieval System

In this module, the user query is processed and transformed to tf-idf vector and applied cosine similarity with the SVD model to obtain the top 10 articles that are contextly similar.

Code

```
def retrieve_similar_articles(query):
    query_tokens = generate_tokens(query)
    query_tokens = capture_lemmatization(query_tokens)
    query_res = ' '.join(query_tokens)
    query_vec = vectorizer.transform([query_res])
    query_lsi = svd_modeling.transform(query_vec)
    sims = cosine_similarity(query_lsi, lsa)
    sims = [(i, sim) for i, sim in enumerate(sims[0])]
    sims = sorted(sims, key=lambda item: -item[1])
    print("\n")
    print(sims[:10])
    print("\n")
    for sim in sims[:10]:
      print(json.dumps(articles[sim[0]]['heading']))
    print("\n")
    return sims
```

Output

Enter the query: Assam Class 10 board exam paper [(3, 0.7100767920542688), (8623, 0.5194617332017587), (9121, 0.46800444210729236), (9578, 0.44447625970847027), (9569, 0.4421496142747494), (9740, 0.4407045110575211), (9650, 0.37280291938781335), (8212, 0.3667350533146309), (4983, 0.35336727716884364), (3862, 0.3488305230265929)]
Closest articles: "Assam Class 10 board exam paper leak govt\u2019s failure: CM Himanta Biswa Sarma" "Among 42 held in Mansa, three are Class 12 students, have Board exams on March 24" "Class 12 student caught \u2018using mobile phone during board exam\u2019" "Class 10 student apprehended in theft case allowed to write board exams" "Karnataka HC permits government to conduct board exams for students of Classes 5 and 8" "Karnataka govt challenges high court order quashing board exams for classes 5 and 8" "Gang busted, 5 held for leaking NTRO exam paper" "Hours before Board exam, Punjab Police lets go five students detained on Sunday" "IPL 2023: Ravi Bishnoi reveals he skipped Class 12th board exams to be Rajasthan Royals net bowler" "The bulldozer fix: Ashok Gehlot govt now uses it as exam paper leaks rock its boat"

CHAPTER 6
TESTCASES AND PERFORMANCE METRICS

6.1 India Genre

Enter the query: Assam class 10

Cosine similarity values ranked

[(3, 0.4726626259393286), (11423, 0.3370870072432971), (8623, 0.26306253195595414), (9196, 0.2625963175483801), (8609, 0.2505581114341634), (3233, 0.24985795526879592), (9507, 0.23995665927803467), (9121, 0.23643292481115324), (4585, 0.23062402099507606), (9299, 0.23045826477910095)]

Relevant articles

"Assam Class 10 board exam paper leak govt\u2019s failure: CM Himanta Biswa Sarma"

"Babies in first class: Which side of the aisle are you on?"

"Among 42 held in Mansa, three are Class 12 students, have Board exams on March 24"

"After singer gaffe, now cheques given to winners bounce in Assam"

"Class 6 student gangraped in Malda school, 3 held: Police"

"Via Ramcharitmanas row, Akhilesh\u2019s message to most backward classes"

"Class 12 student dies by suicide in Gurgaon society"

"Class 12 student caught \u2018using mobile phone during board exam\u2019"

"The rise of Lachit Borphukan as \u2018Hindu warrior\u2019 and \u2018Assam\u2019s Shivaji\u2019"

"Assam question paper leak: 3 arrested, 22 detained"

Performance metrics

Confusion Matrix:

[[0. 0.]

[5. 5.]]

Accuracy: 0.5

Precision: 1.0

Recall: 0.5

F1 Score: 0.6666666666666666

6.2 Sports Genre

Enter the query: will ms dhoni be the next csk captain?

Cosine similarity values ranked

[(5459, 0.646847161185902), (4999, 0.4405113410028404), (5765, 0.3720208380454616), (6418, 0.3570380535020649), (7046, 0.35335594217938626), (7155, 0.3434150240613335), (5190, 0.3400710370867735), (7162, 0.29066533753123786), (6350, 0.26158838161448894), (5293, 0.2588017909957046)]

Relevant articles

"MS Dhoni can play the IPL next year as well: Suresh Raina on the CSK captain"

"IPL 2023: CSK spots MS Dhoni\u2019s successor"

"Will MS Dhoni pass the baton to Ben Stokes in what could be his final season for CSK?"
"\u2018I\u2019m not going to be MS Dhoni as a captain\u2019: RCB\u2019s Faf du Plessis ahead of IPL 2023"
"Former India captains Sourav Ganguly and MS Dhoni meet each other"
"\u2018It\u2019s going to be awkward\u2019: Captains on WPL distraction"
"CSK\u2019s 2018 triumph tells you about leadership: Sunil Gavaskar praises MS Dhoni"
"I played for MS Dhoni, then I played for the country: Suresh Raina talks about bond with former India captain"
"I was always MS Dhoni\u2019s right-hand man: Virat Kohli"
"IPL 2023: Will Gujarat Titans include Kane Williamson for absent David Miller? Who will be CSK \u2018keeper if MS Dhoni sits out?"

Performance metrics

Confusion Matrix:

[[0. 0.]
 [0. 10.]]

Accuracy: 1.0

Precision: 1.0

Recall: 1.0

F1 Score: 1.0

6.3 Technology Genre

Enter the query: Artificial Intelligence and its trends

Cosine similarity values ranked

[(11668, 0.7503012682136081), (8834, 0.2620665340158651), (11426, 0.2352316250603264), (11723, 0.2307742382137832), (1383, 0.22631439295286582), (11655, 0.22538646578692262), (10239, 0.2190433847929068), (11795, 0.2145358227853255), (3474, 0.21413472798056257), (9785, 0.2085010764026469)]

Relevant articles

"Can ChatGPT-led artificial intelligence detect Alzheimer\u2019s early on?"

"\u2018Why hasn\u2019t Amritpal been arrested yet?\u2019 HC slams Punjab govt over \u2018intelligence failure\u2019"

"Wellness trends to look forward to in 2023"

"Nail slugging: Know more about this TikTok trend"

"Cropin looks to cater to agri sector\u2019s growing digitisation and predictive intelligence demands"

"Popular fashion trends that faded away in 2022"

"Study links this artificial sweetener to blood clots, stroke, heart attack and death; know more"

"Home d\u00e9cor: Top furniture trends to look out for in 2023"

"On Pulwama attack anniversary, Digvijaya Singh sparks off fresh row with \u2018blatant intelligence failure\u2019 charge"

"Dampening effect of artificial sweetener on immune response could help treat autoimmune disease: Study"

Performance Metrices

Confusion Matrix:

[[0. 0.]

[8. 2.]]

Accuracy: 0.2

Precision: 1.0

Recall: 0.2

F1 Score: 0.33333333333333337

6.4 Entertainment Genre

Enter the query: Ranbir Kapoor's new film

Cosine similarity values ranked

[(10801, 0.629460324543815), (12093, 0.48695168749278783), (11729, 0.4800970041108088), (11298, 0.4432516945397428), (10528, 0.41460873789849756), (11935, 0.2992475064402437), (7427, 0.24236444873913834), (11646, 0.21590733697687292), (11653, 0.19646173231533534), (11991, 0.19303823069276213)]

Relevant articles

"Watch: Alia Bhatt impresses with her cardio moves as she grooves to Ranbir Kapoor\u2019s new song"

"Ranbir Kapoor, Mahira Khan, Hrithik Roshan attend the Red Sea Film Festival in striking looks"

"\u2018Was always that guy who used to cancel workouts. But now, I feel guilty\u2019: Ranbir Kapoor"

"Airport fashion: Ranbir Kapoor to Suhana Khan, celebs keep it comfy and stylish"

"Ranbir Kapoor\u2019s fitness trainer shares secret behind his physical transformation for \u2018Animal\u2019"

"Sonam Kapoor, Priyanka Chopra, Kareena Kapoor Khan and others spell sartorial magic at Red Sea Film Festival"

"Short film by KEM Hospital bags first prize at Maha Arogya film festival"

"Mira Kapoor, Anshula Kapoor enjoy \u2018gajar ka halwa\u2019; here\u2019s a recipe you can try"

"\u2018As actors, we are scrutinised for films and also on a personal level; this can negatively impact mental health\u2019: Janhvi Kapoor"

"Screenshots have generated new forms of storytelling, from Twitter fan fiction to desktop film"

Performance Metrices

Confusion Matrix:

[[0. 0.]

[4. 6.]]

Accuracy: 0.6

Precision: 1.0

Recall: 0.6

F1 Score: 0.7499999999999999

CHAPTER 7
CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

In conclusion, the retrieval of semantically relevant documents using Latent Semantic Analysis (LSA) holds great promise for improving the accuracy and effectiveness of information retrieval systems. By considering the underlying semantic meaning and context of queries and documents, LSA-based systems can provide more relevant and meaningful search results compared to traditional keyword-based approaches. The challenges associated with building such systems, including corpus creation, efficient handling of large data volumes, and addressing complexities in linguistic contexts, require careful consideration and robust implementation. However, the potential benefits of LSA in enhancing document retrieval justify the efforts invested in overcoming these challenges.

7.2 FUTURE WORKS

Moving forward, several avenues for future work exist in the domain of retrieval systems utilizing LSA. Firstly, further research and development can focus on improving the efficiency and scalability of LSA algorithms to handle increasingly large and diverse datasets. This can involve exploring distributed computing frameworks and parallel processing techniques. Additionally, refining the preprocessing techniques to better capture and represent the semantic relationships within the text can enhance the accuracy of LSA-based retrieval systems.

Furthermore, integrating other advanced natural language processing techniques, such as deep learning models and entity recognition, with LSA can augment the semantic understanding of documents and queries. This can lead to more nuanced and contextually aware retrieval results. Additionally, incorporating user feedback and relevance feedback mechanisms into LSA-based retrieval systems can further personalize and refine the search results based on user preferences and relevance judgments.

Moreover, evaluating and benchmarking the performance of LSA-based retrieval systems against other state-of-the-art approaches can provide insights into their comparative strengths and weaknesses. This can guide the development of hybrid models that combine the strengths of different techniques to achieve even higher retrieval accuracy and user satisfaction.

In conclusion, the future of retrieval systems focused on semantically relevant document retrieval using LSA is promising. Continued research, innovation, and collaboration across academia and industry can further advance the capabilities of these systems, ultimately enhancing the user experience and enabling more efficient and effective access to relevant information in the digital age.

REFERENCES

1. Merrouni, Z. A., Frikh, B., & Ouhbi, B. (2019). Toward Contextual Information Retrieval: A Review And Trends. Procedia Computer Science. https://doi.org/10.1016/j.procs.2019.01.036
2. C. Wenli, "Application Research on Latent Semantic Analysis for Information Retrieval," 2016 Eighth International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Macau, China, 2016, pp. 118-121, doi: 10.1109/ICMTMA.2016.37.
3. A. N. K. Zaman, P. Matsakis and C. Brown, "Evaluation of stop word lists in text retrieval using Latent Semantic Indexing," 2011 Sixth International Conference on Digital Information Management, Melbourne, VIC, Australia, 2011, pp. 133-136, doi: 10.1109/ICDIM.2011.6093315.
4. N. Kumar, S. K. Yadav and D. S. Yadav, "Similarity Measure Approaches Applied in Text Document Clustering for Information Retrieval," 2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC), Waknaghat, India, 2020, pp. 88-92, doi: 10.1109/PDGC50313.2020.9315851.
5. W. Sun, L. Zhang, K. Chang and S. Yu, "DSMN: A Personalized Information Retrieval Algorithm Based on Improved DSSM," 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 2021, pp. 1-7, doi: 10.1109/IJCNN52387.2021.9533630.
6. M. Khari, A. Jain, S. Vij and M. Kumar, "Analysis of various information retrieval models," 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2016, pp. 2176-2181.