Web Page Ranking Based on Text Content and Link Information Using Data Mining Techniques

Esraa Q. Naamha and Matheel E. Abdulmunim

Department of Computer Science, University of Technology-Iraq,
Baghdad, Iraq

Abstract–**Thanks to the rapid expansion of the Internet, anyone can now access a vast array of information online. However, as the volume of web content continues to grow exponentially, search engines face challenges in delivering relevant results. Early search engines primarily relied on the words or phrases found within web pages to index and rank them. While this approach had its merits, it often resulted in irrelevant or inaccurate results. To address this issue, more advanced search engines began incorporating the hyperlink structures of web pages to help determine their relevance. While this method improved retrieval accuracy to some extent, it still had limitations, as it did not consider the actual content of web pages. The objective of the work is to enhance Web Information Retrieval methods by leveraging three key components: text content analysis, link analysis, and log file analysis. By integrating insights from these multiple data sources, the goal is to achieve a more accurate and effective ranking of relevant web pages in the retrieved document set, ultimately enhancing the user experience and delivering more precise search results the proposed system was tested with both multi-word and single-word queries, and the results were evaluated using metrics such as relative recall, precision, and F-measure. When compared to Google's PageRank algorithm, the proposed system demonstrated superior performance, achieving an 81% mean average precision, 56% average relative recall, and a 66% F-measure.**

***Index Terms*—Information retrieval, JSON API, Programmable (CSE), Search engine, World Wide Web, Web page ranking.**

I. Introduction

The World Wide Web (WWW) is evolving rapidly as a dynamic, explosive, diverse, vast, and unstructured data repository. It currently serves as an extensive knowledge reference, but it poses several challenges. Web pages are semi-structured, the web is vast, and the meaning of web information varies, affecting the quality of extracted knowledge. A comprehensive understanding and analysis of the web's data structure are essential for efficient IR. Web mining methods, including IR, Natural Language Processing (NLP), Machine Learning (ML), and Database (DB) techniques, address these challenges. To access information on the WWW, individuals utilize search engines such as Bing, Web Crawler, Iwon, Yahoo, Google, and similar platforms. These search engines consist of three main components: a crawler (spider or robot) that navigates the web and captures pages, an indexing module that parses downloaded pages, constructs an index using keywords, and stores Uniform Resource Locators (URLs) of relevant pages. When users enter keywords into a search engine's interface, the query processor compares these keywords to the index and provides users with a list of relevant pages. However, before displaying the results, search engines employ a ranking mechanism to prioritize the most relevant pages at the top and the least relevant ones at the bottom (Sharma, Yadav and Garg, 2020; Alhaidari, Alwarthan and Alamoudi, 2020). The use of web search engines has become immensely popular for efficiently finding valuable information (Sharma, et al., 2019; Guwta, 2021). These engines can be divided into two generations based on their indexing techniques (Ali and Khusro, 2021). In the early days of the web, first-generation search engines relied solely on index terms extracted from web page content. Consequently, web page searches followed a conventional document retrieval approach. However, the unique characteristics of web pages, such as their hyperlink structures and vast quantity, posed challenges for efficient searching. Consequently, users often found the retrieval accuracy and usability of these search engines unsatisfactory (Mustafa, et al., 2022; Afolabi, Makinde and Oladipupo, 2019; Phyu and Thu, 2021). First-generation search engines did not fully leverage the unique features of web pages. Second-generation search engines addressed these issues by considering the hyperlink structures associated with web pages. For instance, techniques such as Hypertext-Induced Topic Search (HITS) and Page Rank (PR) utilize web pages' hyperlink structures. Compared to first-generation search engines, algorithms like these achieve higher retrieval accuracy by assigning weights to web pages based on their hyperlink structures. However, these algorithms have a limitation in that they primarily assess the importance of web pages without taking into account the relative significance of the content between hyperlinked pages (Tyagi and Gupta,

ARO-The Scientific Journal of Koya University
Vol. XII, No. 1 (2024), Article ID: ARO.11397. 12 pages
Doi: 10.14500/aro.11397
Received: 11 September 2023; Accepted: 01 February 2024
Regular research paper: Published: 15 February 2024
Corresponding author's e-mail: cs.20.29@grad.uotechnology.edu.iq

2018; S. and A., 2019; Payal, 2020). As a result, the problem of irrelevant web pages ranking highly in response to a user's query persists. Therefore, a technique must be developed to accurately represent web page contents and provide users with relevant results (Sharma, Yadav and Thakur, 2022; Team, et al., 2023).

The aim of this study is to propose a Web IR approach that could incorporate the analysis of the page's text content, link information, and log files to rank relevant Web pages higher in the retrieved document set. By incorporating text content analysis, link analysis, and log file analysis, this approach seeks to leverage multiple component to gain a deeper understanding of Web pages, user behavior, and relevance. Each component provides valuable insights that, when integrated, can enhance the accuracy and effectiveness of Web IR. This enables the algorithm to rank relevant Web pages higher in the retrieved document set, resulting in an improved user experience and more accurate search results.

Robert and Brown (2004) introduced the PR algorithm for ranking web pages. The web consists of a complex structure of interlinked web pages, and PR calculates page rankings based on their link structures. When a page has important incoming links, the PR algorithm also considers the pages it connects to as important. PR spreads ranking influence through backlinks, assigning a high rank to a page if the sum of its backlink ranks is substantial.

Xing and Ghorbani (2004) introduced the Weighted PageRank (WPR) algorithm as an enhanced version of the PR algorithm. This system evaluates a page's popularity, determines its page rank, and takes into account the relevance of both outgoing and incoming links. The number of outgoing and incoming links a page has influences its popularity. Unlike the PR algorithm, the WPR algorithm does not evenly distribute a page's rank among its outgoing links. Instead, it assigns weight values to both inbound and outbound links based on their importance.

Kleinberg (2011) suggested the HITS algorithm, which classifies web pages into two categories known as Hubs and Authorities. Authorities are pages that contain essential content and are linked to by numerous hyperlinks. This algorithm determines a web page's rank by analyzing its content in relation to a given query. Moreover, the HITS algorithm relies on the web's structure after collecting web pages.

This paper provides the following main contributions:

- A Google programmable Customized Search Engine (CSE) is created and implemented to extract links and their associated metadata from web pages in the Google database and save it in JavaScript Object Notation (JSON) format using the Google Application Programming Interface (API), where a web site offers a set of structured Hypertext Transfer Protocol (HTTP) requests that return JSON files.
- An innovative ranking method has been proposed to re-rank links retrieved by Google using semantic metadata analysis. This method takes into account several factors, such as the number of links visited across different time periods, regions, and related topics and queries. By incorporating these elements, a novel ranking algorithm has been developed to deliver more accurate and relevant search results.
- The top link from each semantic metadata criterion serves as the root web page or seed URL for the web crawling algorithm. The goal of this step is to initiate the extraction of hyperlinks or URLs from web pages, saving them in a local database for further analysis and retrieval purposes. By starting the web crawling process with seed URLs derived from the top links of the semantic metadata criteria, the approach focuses the crawling effort on specific web pages that are likely to be more relevant or representative based on these criteria. This allows for a targeted collection of URLs, which can then be used for subsequent retrieval and analysis tasks.
- The Singular Value Decomposition (SVD) algorithm has been improved by selecting the top-k dominant features from the global feature vector and retaining 95% of the energy to determine which top-k features to consider based on the eigenvalues of the words or features. This transformation reduces a larger, high-dimensional, sparse matrix to a smaller, more manageable one.
- A new similarity measure is employed to assess the similarity between documents and a user's query. This measure takes into account multiple aspects, including syntactic, semantic, and sentiment-related similarities. By considering these various dimensions, a more comprehensive understanding of document similarity is achieved. This holistic understanding enables the ranking of documents based on their relevance to the user's query, considering multiple dimensions of similarity.

II. Background

A. Vector Space Model (VSM)

After preprocessing, handling document complexity involves transforming the resulting documents from complete text representations into document vectors that describe their contents. This vector space IR system is known as VCM. One key advantage of this representation is its ability to leverage the algebraic structure of the vector space. VCM facilitates information filtering, IR implementation, indexing, and ranking of information relevance. Using vectors in natural language, documents are represented as vectors of index terms in a multidimensional linear space. VCM possesses several attractive properties, some of which are listed below (Allahyari, et al., 2017; Shahmirzadi, Lugowski and Younge, 2019):

- It can handle heterogeneous document formats.
- It can process various types of multimedia data.
- It can work with documents in multiple languages.
- The IR process can be fully automated.
- Computational work can be performed entirely during the preprocessing stage, enabling real-time query processing.

B. Term Weighting Schemes

Clarifying term weighting involves balancing inclusiveness and accuracy in the search process, where inclusiveness relates to recall and accuracy relates to specificity. Term weighting schemes impact the performance of VSMs, which are the functions that determine the component vectors (Jain, Vishwakarma and Jain, 2023). The performance of the VSM can be significantly enhanced through appropriate term weighting. Initially, single-term statistics were employed to weight the VSM. Term weighting involves three primary factors, with local weights assigned based on the number of term appearances within a document (Jain, Jain and Vishwakarma, 2020; Rathi and Mustafi, 2023).

Because a term that appears ten times in a document may not necessarily be 10 times as relevant as a term that appears only once in the same document, Logarithmic Term Frequency (LTF) is introduced as a local weight to adjust within-document frequency. The following equation can be employed to calculate LTF (Wu and Gu, 2014; Nassar, Kanaan and Awad, 2010):

$$LTF\left(L_{ij}\right) = Log\left(f_{ij} + 1\right) \tag{1}$$

Global weights are determined by assessing the occurrences of each term across the entire collection. Global Frequency Inverse Document Frequency (GFIDF) is often the most effective global weight. In cases where a term is present in every document or occurs only once in a single document, its weight is set to one, the minimum weight possible. Terms that occur frequently are assigned a significant weight, which is proportional to the number of documents they appear in. The GFIDF equation can be calculated as follows (Wu and Gu, 2014; Nassar, Kanaan and Awad, 2010):

$$GFIDF\left(G_i\right) = \frac{\sum_{j=1}^{n} f_{ij}}{\sum_{j=1}^{n} x\left(f_{ij}\right)} \tag{2}$$

The normalization factor takes into account differences in document lengths. Among the commonly used normalization methods in the VSM, Cosine Normalization (CN) is prominent. CN divides by the weighted document vector to ensure that the magnitude of these vectors becomes one. This method allows us to examine the angle separating the weighted vectors. Longer documents receive reduced individual term weights, which favors retrieval for shorter documents over longer ones. The CN equation can be calculated as follows (Wu and Gu, 2014; Nassar, Kanaan and Awad, 2010):

$$CN(N_j) = \left[\sum_{i=1}^{m}\left(g_i * t_{ij}\right)^2\right]^{-\frac{1}{2}} \tag{3}$$

C. Latent Semantic Indexing (LSI)

When implemented, this technique compresses document vectors into a lower-dimensional space characterized by dimensions obtained from co-occurrence patterns. LSI determines the structure of relationships between words and documents by analyzing word co-occurrence patterns. This process organizes data into a semantic arrangement that maximizes the advantages of implicit higher-level associations between text objects and words. Furthermore, it addresses challenges arising from polysemy (words with diverse meanings) and synonymy (multiple words representing the same concept) within efficient IR (Al-Anzi and Abuzeina, 2020). The SVD approach retains the most relevant distance information by reducing the dimensionality of document vectors. This reduction results in some information loss and the overlapping of content words. However, this information loss can have a positive aspect. According to P. P., (2020), loss represents noise in the initial term-document matrix, revealing latent similarities in the document collection. SVD, an orthogonal decomposition, is employed to compute the rank-t approximation where (t<*min* (*m*,*n*)) in a m × n matrix A once it has been properly constructed and weighted. As stated by Qi, Hessen and van der Heijden (2023), the original matrix A is decomposed into three new matrices—S, U, and V using the SVD method.

$$A = USV^T \tag{4}$$

The diagonal elements of S, which monotonically decrease in value and are known as the singular values of matrix A, are represented by the left and right singular vectors in the columns of U and V, respectively.

D. Similarity Measure

The measurement of similarity between a query and a document in a document collection is a crucial component of the IR system. This similarity is mathematically quantified, with the higher values indicating greater likeness. Typically, non-negative values within the range of [0,1] are used to represent similarity measures. A value of 1 indicates complete similarity, while a value of 0 indicates no similarity (Reddy, et al., 2018; Ghani and Hussain, 2021; Wang and Dong, 2020). There are many similarity measures, including cosine similarity, Jensen Shannon Divergence (JSD) similarity, sentiment similarity, and more.

A widely favored measure in IR models is cosine similarity. In this approach, queries and documents are treated as vectors in term space, allowing for easy computation through vector operations. Cosine similarity is defined as (Thakur, et al., 2019):

$$sim_{cosine} = \frac{\sum_{k=1}^{n} w_{kj} \cdot w_{kq}}{\sqrt{\sum_{k=1}^{n} w_{kj}^{2}} \sqrt{\sum_{k=1}^{n} w_{kq}^{2}}} \tag{5}$$

JSD is a metric that quantifies the distance between two probability distributions, indicating their degree of dissimilarity or similarity. It is built on the foundation of Kullback-Leibler divergence, which is used to compare two probability distributions. The JSD between two probability distributions, Q and P, is defined as (Lu, Henchion and Namee, 2020):

$$JSD(P \| Q) = (KL(P \| M) + KL(Q \| M)) / 2 \tag{6}$$

Where M is the average distribution, defined as:

$$M = (P + Q) / 2 \tag{7}$$

In addition, $KL(P\|Q)$ is the Kullback-Leibler divergence between Q and P, specified as:

$$KL(P\|Q)=\sum p(i)*log(p(i)/q(i)) \quad (8)$$

The sum is taken over all possible values of i, where $q(i)$ and $p(i)$ represent the probabilities of i in the distributions Q and P, respectively.

According to Zheng and Fang (2010), sentiment similarity in the context of IR quantifies the similarity between queries or documents based on their emotional or sentimental content. A Python library for NLP known as TextBlob actively utilizes the Natural Language Toolkit (NLTK) to accomplish its tasks. NLTK is a library that simplifies user access to various lexical resources and enables tasks such as classification and categorization (Hazarika, Konwar and Bora, 2020).

III. Proposed System

The concept behind the proposed system relies on various measures to enhance result quality, avoiding reliance on a single measure. In this regard, diverse actions are taken for web page ranking, incorporating content, structure, and log data. The proposed system is implemented using the Python 3 language. The proposed system consists of three stages: dataset collection (link's metadata scraping), semantic metadata analysis, and Web IR. The framework of the proposed system is illustrated in Fig. 1. An overview of the proposed system for each level is provided in the following steps:

A. The first stage of the proposed system is dataset collection (link's metadata scraping), in which a Google programmable CSE is created and implemented to extract links and their

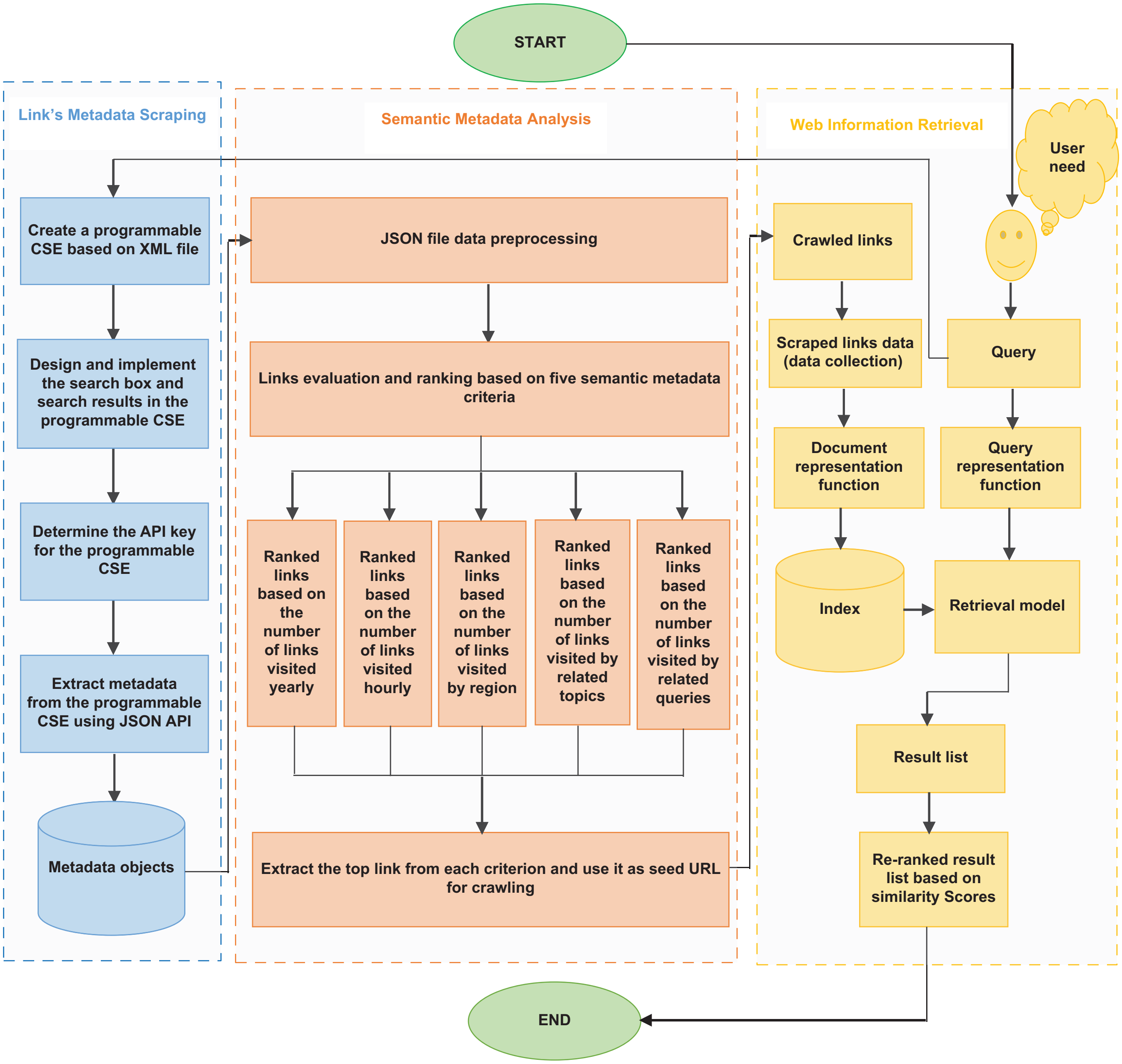
START
Link's Metadata Scraping
Create a programmable CSE based on XML file
Design and implement the search box and search results in the programmable CSE
Determine the API key for the programmable CSE
Extract metadata from the programmable CSE using JSON API
Metadata objects
Semantic Metadata Analysis
JSON file data preprocessing
Links evaluation and ranking based on five semantic metadata criteria
Ranked links based on the number of links visited yearly
Ranked links based on the number of links visited hourly
Ranked links based on the number of links visited by region
Ranked links based on the number of links visited by related topics
Ranked links based on the number of links visited by related queries
Extract the top link from each criterion and use it as seed URL for crawling
Web Information Retrieval
User need
Crawled links
Scraped links data (data collection)
Query
Document representation function
Query representation function
Index
Retrieval model
Result list
Re-ranked result list based on similarity Scores
END

Fig. 1. Framework of the proposed system.

associated metadata from web pages in the Google database and save it in a JSON format using the Google API, where a web site offers a set of structured HTTP requests that return JSON files. The following are the details:

1. The first step involves creating a basic search engine using a programmable search engine control panel. This control panel allows users to configure and customize the search engine to meet their specific requirements. Once the search engine is set up, the control panel's overview page provides access to download annotations and context XML files. These files offer enhanced control, flexibility, and access to advanced features of the programmable search engine."
2. Once the programmable CSE has been created and configured, the next step involves integrating programmable search elements into an HTML file. This includes adding both the search box and search results to the desired location on the web site where the search engine will be embedded.
3. Afterward, the API is utilized, playing a crucial role in enabling communication between the programmable CSE and the service provider, in this case, Google. It serves as the interface through which users can interact with the programmable CSE and retrieve the desired search results and metadata. To use the API for the programmable CSE, both the search engine ID and API key should be identified and obtained initially. These are essential components that provide access to the search engine's functionalities and authenticate API requests.
4. Finally, the Custom Search JSON API is one of the powerful tools that enable developers to programmatically retrieve search results from a programmable CSE and display them on their applications or web sites. This API allows developers to make RESTful requests to obtain search results, both for web and image searches, in JSON format.

B. The second stage of the proposed system, semantic metadata analysis, is a crucial step towards improving the ranking of the links extracted by Google in the first stage. By leveraging semantic metadata criteria, the new ranking approach aims to assess the significance of these links across various dimensions such as time periods, regions, and related queries and topics. The following provides more details:

1. The first step involves pre-processing JSON file data. The JSON file is opened in write mode, and a Python dictionary is serialized as a JSON formatted stream to the opened file using the json.dump() function. This function should be set to ASCII = False if the JSON data contain non-ASCII characters. Subsequently, the json.load() function automatically returns a Python dictionary, making it easier to work with the JSON data.
2. The next step involves re-ranking the links retrieved by Google based on semantic metadata criteria. This process helps prioritize more relevant and significant links for a given query. After the re-ranking is completed, the top 10 links with the highest hits from each criterion's results are extracted and presented to users as the most relevant and significant results for their query.

C. The third stage of the proposed system, Web IR, is a critical step in further refining the ranking of relevant pages from the retrieved document set. In this stage, data mining techniques are employed to analyze both the text content and link information of the web pages to ascertain their relevance and importance. The following provides more details:

1. The first step is web page crawling, which is crucial for gathering the relevant data needed for further processing and analysis. This process involves extracting the top link from each semantic metadata criterion and using them as seed URLs in a web crawling algorithm. The objective is to retrieve all the hyperlinks or URLs present within the content of those web pages.
2. The next step involves collecting text data from each crawled link using BeautifulSoup, a powerful Python library for web scraping and parsing HTML or XML documents. Once the relevant text data are extracted, it will be saved into a local database for further analysis and processing.
3. Subsequently, the necessity for data cleansing becomes evident to generate meaningful results. In essence, meaningful terms must be extracted from the text through a series of pre-processing actions. During the preprocessing stage, tasks such as tokenization, removal of stop words, and stemming are performed.
4. Using a statistically based VCM, a document is theoretically represented as a vector of keywords, with associated weights indicating the significance of keywords within the document and across the entire document collection. Similarly, a query is represented as a list of keywords with corresponding weights, signifying the importance of keywords in the query. A term's weight in a document vector can be calculated by combining weighting algorithms for local, global, and normalization. Once term weights are determined, a document-term matrix is constructed, with content words in columns and documents in rows.
5. However, when dealing with a complex document database, the number of terms involved is often substantial. This increased dimensionality poses the challenge of inefficient calculations. Furthermore, the higher dimensionality leads to exceedingly sparse vectors and complicates the identification and utilization of relationships between terms. To address these issues, the use of LSI comes into play. This technique employs the SVD approach to effectively reduce the dimensions of the document-term matrix, facilitating analysis.
6. The final step in the IR process involves developing a ranking function that measures the similarity between query and document vectors. The proposed new similarity considers semantic, syntactic, and sentiment-related aspects to evaluate the relevance of the documents to the user's query. These similarities are then combined to obtain a final quantitative value, indicating how similar the documents are to the user's query. Based on this value, the documents are ranked accordingly.

IV. Experimental Results

This section provides an overview of the stages involved in implementing the proposed system, which include link's metadata scraping results, semantic metadata analysis results, and Web IR results.

A. First Stage Experimental Results: Link's Metadata Scraping

In this stage, the user interacts with the system by entering a search query, which in this case is "information retrieval." The system takes this query as input and proceeds to fetch relevant search results. It imports a JSON file containing links and their associated metadata, which may include page titles, keywords, descriptions, URLs, and other relevant information about the web pages. The system then creates a list of search results that are most pertinent to the user's query, based on metadata analysis and relevance evaluation. These results may be presented as a list of links, accompanied by additional information such as page titles and descriptions to help the user identify the content of each link, as illustrated in Table I.

B. Second Stage Experimental Results: Semantic Metadata Analysis

In this stage, following the creation and implementation of a programmable CSE to extract links and their associated metadata from web pages and retrieve relevant links ranked using Google's PageRank algorithm, as displayed in Table I, the links are re-ranked based on five semantic metadata criteria. The links are independently evaluated for each of the five criteria, considering their hits or occurrences related to each category. For each criterion, the top 10 links with the highest hits are selected as the most significant and relevant links for that specific criterion. The following presents the results for each of the five semantic metadata criteria:

- Criteria 1: Links are ranked based on the number of links visited yearly, which indicates the popularity and traffic of each link over the course of a year, as shown in Table II.
- Criteria 2: Links are ranked based on the number of links visited hourly, reflecting the current popularity and recent traffic patterns for each link, as shown in Table III.
- Criteria 3: Links are ranked based on the number of links visited by region, assessing the popularity of links in specific geographic locations or their relevance to different audiences in different regions, as shown in Table IV.
- Criteria 4: Links are ranked based on the number of links visited by related keyword topics. This criterion may involve identifying links that are frequently visited in the context of related keyword topics to the user's query, as shown in Table V.
- Criteria 5: Links are ranked based on the number of links visited by related search queries, identifying links that are frequently visited in the context of searches similar to the user's query, as shown in Table VI.

C. Third Stage Experimental Results: Web IR

In this stage, after re-ranking the links using the five semantic metadata criteria, the top-ranked link from each criterion is

TABLE I
The List of Relevance Search Results to the User's Search Query, "Information Retrieval"

Rank	Metadata		
	URL	Title	Description
1	https://en.wikipedia.org/wiki/Information_retrieval	Information retrieval.	Resources from a collection of information system resources which are relevant to a particular information demand.
2	https://www.geeksforgeeks.org/what-is-information-retrieval/	What is information retrieval?	A software program which stores, organizes, retrieves, and evaluates information from document repositories, especially textual information, is known as information retrieval (IR).
3	https://nlp.stanford.edu/IR-book/information-retrieval-book.html	Introduction to information retrieval.	The purpose of the book is to present a contemporary computer science perspective on IR. It depends on a course taught at the University of Stuttgart, Stanford University, and the University of Munich in a variety of formats.
4	https://www.engati.com/glossary/information-retrieval	What are the three classic models in information retrieval systems?	The three types of IR models are the non-classical IR model, the alternative IR model, and the classical IR model.
5	https://www.coveo.com/blog/information-retrieval/	The three parts of any information retrieval system.	IR systems serve as a link between users and data repositories. Querying, indexing, and presentation are, at a high level, the three major components of IR system.
6	https://www.upgrad.com/blog/information-retrieval-system-explained/	Information retrieval system explained: types, comparison and components.	An IR system is a collection of algorithms which makes it easier for documents to be shown that are relevant to searches. Simply put, it works to sort and rank documents according to user queries.
7	https://www.librarianshipstudies.com/2020/02/information-retrieval.html	Information retrieval models.	Searching for, locating, and obtaining recorded data and information from a file or database is referred to as IR.
8	https://www.sciencedirect.com/topics/computer-science/information-retrieval-systems	Information retrieval systems.	A conventional IR system experiment consists of the following elements: a set of documents, an indexing system, a predetermined set of queries, and assessment standards.
9	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3137130/	An introduction to information retrieval.	The area of computer science known as IR is concerned with processing documents which contain free text so that they may be quickly retrieved depending on keywords entered by a user.
10	https://paperswithcode.com/task/information-retrieval	Information retrieval progress.	Ranking a list of documents or search results in response to a query is the work of IR.

TABLE II
Web Page Ranking Based On Interest Over Time

Rank	URL	Hits
1	https://www.geeksforgeeks.org/what-is-information-retrieval/	100
2	https://en.wikipedia.org/wiki/Information_retrieval	74
3	https://www.upgrad.com/blog/information-retrieval-system-explained/	40
4	https://www.coveo.com/blog/information-retrieval/	26
5	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3137130/	20
6	https://paperswithcode.com/task/information-retrieval	18
7	https://nlp.stanford.edu/IR-book/information-retrieval-book.html	14
8	https://www.engati.com/glossary/information-retrieval	12
9	https://www.librarianshipstudies.com/2020/02/information-retrieval.html	11
10	https://www.sciencedirect.com/topics/computer-science/information-retrieval-systems	10

TABLE III
Web Page Ranking Based on Hourly Historical Interest

Rank	URL	Hits
1	https://en.wikipedia.org/wiki/Information_retrieval	73
2	https://www.upgrad.com/blog/information-retrieval-system-explained/	70
3	https://www.librarianshipstudies.com/2020/02/information-retrieval.html	69
4	https://paperswithcode.com/task/information-retrieval	68
5	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3137130/	63
6	https://www.geeksforgeeks.org/what-is-information-retrieval/	52
7	https://www.coveo.com/blog/information-retrieval/	44
8	https://www.engati.com/glossary/information-retrieval	42
9	https://nlp.stanford.edu/IR-book/information-retrieval-book.html	34
10	https://www.sciencedirect.com/topics/computer-science/information-retrieval-systems	28

TABLE IV
Web Page Ranking based on Interest by Region

Rank	URL	Hits
1	https://paperswithcode.com/task/information-retrieval	100
2	https://www.upgrad.com/blog/information-retrieval-system-explained/	79
3	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3137130/	36
4	https://www.geeksforgeeks.org/what-is-information-retrieval/	31
5	https://www.geeksforgeeks.org/what-is-information-retrieval/	30
6	https://www.coveo.com/blog/information-retrieval/	28
7	https://www.sciencedirect.com/topics/computer-science/information-retrieval-systems	25
8	https://www.librarianshipstudies.com/2020/02/information-retrieval.html	22
9	https://www.upgrad.com/blog/information-retrieval-system-explained/	20
10	https://www.engati.com/glossary/information-retrieval	17

TABLE V
Web Page Ranking Based on Related Topics

Rank	URL	Hits
1	https://www.sciencedirect.com/topics/computer-science/information-retrieval-systems	100
2	https://www.engati.com/glossary/information-retrieval	99
3	https://nlp.stanford.edu/IR-book/information-retrieval-book.html	66
4	https://en.wikipedia.org/wiki/Information_retrieval	30
5	https://www.coveo.com/blog/information-retrieval/	17
6	https://www.upgrad.com/blog/information-retrieval-system-explained/	10
7	https://www.librarianshipstudies.com/2020/02/information-retrieval.html	5
8	https://www.geeksforgeeks.org/what-is-information-retrieval/	4
9	https://paperswithcode.com/task/information-retrieval	3
10	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3137130/	2

TABLE VI
Web Page Ranking Based On Related Search Queries

Rank	URL	Hits
1	https://nlp.stanford.edu/IR-book/information-retrieval-book.html	100
2	https://www.geeksforgeeks.org/what-is-information-retrieval/	71
3	https://www.sciencedirect.com/topics/computer-science/information-retrieval-systems	70
4	https://www.librarianshipstudies.com/2020/02/information-retrieval.html	69
5	https://www.upgrad.com/blog/information-retrieval-system-explained/	68
6	https://www.coveo.com/blog/information-retrieval/	57
7	https://en.wikipedia.org/wiki/Information_retrieval	48
8	https://paperswithcode.com/task/information-retrieval	44
9	https://www.engati.com/glossary/information-retrieval	37
10	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3137130/	28

extracted, representing the most significant and relevant link for that criterion. These top-ranked links are then used as seed URLs for crawling. A web crawling algorithm is employed to systematically collect all the links within the content of these web pages. The objective is to explore and discover additional web pages linked from the root web pages, thereby expanding the search space and gathering more information. After extracting each possible link from the top links, the web page text data are collected from each extracted link using the BeautifulSoup scraping tool. Subsequently, the collected text data undergoes preprocessing through several techniques: tokenization, stop word removal, and stemming. At this point, the documents containing only content words are represented as vectors following the VSM of IR. The terms in the query are also represented as a query vector. The content words from the documents are placed in the columns, and the corresponding documents are in the rows of the document-term matrix. The cells in the matrix contain binary values, indicating the presence or absence of terms in the corresponding documents. Terms that occur frequently and those that appear only once are not distinguished by binary weight; instead, term frequency (how often a word or phrase appears in the relevant document) is considered. A term's weight is determined by considering its local, global, and normalization weighting schemes. The resulting document-term matrix is typically high dimensional and sparse due to the large number of terms. To address this, the SVD method, commonly used for LSI, is applied to reduce the dimensionality while retaining dominant and significant features. For example, consider a sample document-term matrix denoted as X in the context of dimensionality reduction using SVD. Table VII shows a document-term matrix with nine documents and ten terms, initially forming the feature set.

Tables VIII-X display the resulting matrix elements obtained through the SVD applied to the document-term matrix X. The SVD factorizes the matrix X into three

TABLE VII
Document-Term Matrix

Documents	Terms (Features)									
	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
D_1	0.0000	0.5695	0.0000	0.0000	0. 4796	0.4055	0.0000	0.0000	0.0000	0.4796
D_2	0.0000	0.0000	0.0000	0.0000	0.0000	0.6931	0.3757	0.3465	0.0000	0.0000
D_3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.3757	0.0000	0.0000	0.0000
D_4	0.0000	0.0000	0.2310	0.0000	0.6931	0.4435	0.3757	0.5695	0.0000	0.4796
D_5	0.0000	0.0000	0.0000	0.5695	0.0000	0.3465	0.0000	0.0000	0.3857	0.0000
D_6	0.6931	0.4435	0.3857	0.0000	0.0000	0.4055	0.0000	0.0000	0.2310	0.0000
D_7	1.0986	0.6931	0.4055	1.0986	0.0000	0.2310	0.0000	0.3857	0.3465	0.0000
D_8	0.4435	0.0000	0.2310	0.4435	0.0000	0.3465	0.0000	0.0000	0.0000	0.0000
D_9	0.4435	0.5695	0.3857	0.5695	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

TABLE VIII
Matrix U (Document×Document Matrix)

−0.1349	−0.2729	−0.1618	−0.4403	0.7357	−0.0484	−0.2167	−0.2213	−0.2186
−0.1525	−0.4215	0.6911	−0.0219	−0.1221	−0.3667	0.3545	−0.0364	0.2114
−0.0137	−0.1428	−0.0284	0.1846	−0.2607	−0.236	−0.2333	−0.7336	−0.4778
−0.1920	−0.7976	−0.3913	0.1966	−0.1377	0.1438	−0.1904	0.2228	0.0982
−0.1642	−0.0298	−0.4653	0.1179	0.332	0.4638	−0.5669	−0.2121	0.2238
−0.3748	0.0532	−0.0049	−0.7739	−0.4096	−0.0031	−0.2974	0.0224	0.0319
−0.7796	0.2643	−0.0444	0.3361	0.1631	−0.3178	−0.0662	0.221	−0.1632
−0.2689	−0.0040	0.1456	0.0412	−0.2158	0.6781	0.4732	−0.0094	−0.4176
−0.2720	0.1330	−0.3204	0.0738	−0.076	0.1047	0.3139	−0.5171	0.6462

TABLE IX
Eigen Value Matrix S

6.3313	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	3.8887	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	2.0543	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	1.735	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	1.5891	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	1.1554	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0549	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.665	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.3829	0.0000

matrices: a diagonal matrix (Eigenvalue matrix), a left singular matrix (document x document matrix), and a right singular matrix (term x term matrix), as shown in Eq. (4):

$$X = [Document\,Matrix]\,x$$
$$[Eigenvalue\,Matrix]\,x\ [Term\,Matrix]^T$$

The diagonal matrix, presented in Table IX, displays a distinct arrangement, with Eigenvalues neatly ordered along the diagonal. These Eigenvalues symbolize the relevance assigned to corresponding terms or features in the initial document-term matrix.

Consider the absolute values in the first column of the term matrix, which, when sorted, represents the terms in order of their relative importance. Term T1 emerges as the most significant, while term T10 is the least significant. By selecting the top-k most significant terms, all inconsequential terms can be removed from the global feature set once the significant terms are identified. The criteria for selecting the top-k terms are to retain 95% of the energy from the Eigen values of the term vectors. The sorted characteristics and their corresponding Eigenvalues are presented in Table XI.

The sum of all Eigenvalues, which equals 18.8566, represents the total energy of the Eigenvalue matrix. The energy for the top-k Eigenvalues, where k can be one of 6, 7, 8, 9, or 10. To retain 95% of the energy, the top 7 terms are considered, leading to a reduction in the dimensions of the initial matrix from 10 terms to 7 terms. After achieving dimension reduction and structuring the information systematically in a semantic format within the document-term matrix, the next step involves introducing a ranking function. This function assesses the similarity between query and document vectors, aiding in document retrieval. In this system, a new similarity measure is proposed, incorporating three different measures: syntactic similarity, semantic similarity, and sentiment similarity, resulting in a more accurate and meaningful document ranking. Syntactic similarity utilizes the cosine similarity measure, which calculates the cosine of the angle between the query and document vectors, as shown in Eq. (9):

$$\text{Syntactic Simalirity}\left(d,q\right) = \text{Coscos}\left(d,q\right) \tag{9}$$

The JSD similarity metric is employed to assess the semantic similarity between the query and document vectors. To measure the semantic relationship between terms in the query and document, JSD calculates the divergence between their probability distributions, as indicated in Eq. (10):

$$\text{Semantic similarity}\left(d,q\right) = \text{JSD}\left(\theta_d,\theta_q\right) \tag{10}$$

To analyze sentiment polarity, TextBlob, a Python library for sentiment analysis, is utilized to determine whether the sentiment expressed in the query matches that of the documents, as shown in Eq. (11):

$$\text{Sentiment Simalirity}\left(d,q\right) = \text{Polarity}\left(d\right) \times \text{Polarity}\left(q\right) \tag{11}$$

After calculating the individual similarity values for each aspect, the final similarity score is determined by combining them using the arithmetic mean. The average similarity is calculated by summing up the similarity scores for syntactic similarity, semantic similarity, and sentiment-related similarity, and then dividing the sum by 3, as shown in Eq. (12):

$$\frac{\begin{array}{c}\text{Syntactic Simalirity+Semantic similarity}\\ \text{+Sentiment Simalirity}\end{array}}{3} \tag{12}$$

TABLE X
Matrix V (Trem x Term Matrix)

0.5732	0.2644	−0.1547	−0.2447	−0.3912	−0.153	−0.0059	0.2726	−0.5145	−0.0462
0.3697	0.1136	−0.2804	−0.2699	0.3627	−0.504	0.0956	−0.4121	0.348	−0.1155
0.2981	−0.0903	−0.2996	−0.0728	−0.4254	0.5242	0.221	−0.0907	0.5108	0.1849
0.4808	0.2294	0.0765	0.7154	0.3332	0.2537	0.0205	−0.1136	−0.0968	−0.0231
0.0819	−0.4804	−0.4597	−0.0272	0.2896	0.207	−0.1555	0.3372	−0.058	−0.5314
0.3505	−0.4191	0.6769	−0.3264	0.1406	0.1585	0.2636	−0.0745	−0.0593	−0.1155
0.0869	−0.5553	−0.0584	0.3203	−0.4143	−0.2726	−0.2461	−0.4879	−0.1829	0.0000
0.1775	−0.2456	0.1243	0.2943	−0.0609	−0.4679	0.0928	0.6126	0.3824	0.2311
0.2083	0.074	0.2025	−0.1844	0.0538	0.1237	−0.8821	0.047	−0.2418	0.1386
0.0516	−0.2753	−0.2692	−0.1405	0.3763	0.0826	0.025	0.0022	−0.3144	0.7625

TABLE XI
Sorted Feature Corresponding Eigen Values

Features before sorting		Features after sorting		Eigen values	
T1	0.5732	T1	0.5732	EV1	6.3313
T2	0.3697	T4	0.4808	EV2	3.8887
T3	0.2981	T2	0.3697	EV3	2.0543
T4	0.4808	T6	0.3505	EV4	1.7350
T5	0.0819	T3	0.2981	EV5	1.5891
T6	0.3505	T9	0.2083	EV6	1.1554
T7	0.0869	T8	0.1775	EV7	1.0549
T8	0.1775	T7	0.0869	EV8	0.6650
T9	0.2083	T5	0.0819	EV9	0.3829
T10	0.0516	T10	0.0516	EV10	0.0000

Subsequently, the documents are ranked in descending order of their final similarity scores. Documents with higher similarity scores are given higher priority in search results, as they are considered more relevant to the user's query, as shown in Table XII.

V. Evaluation and Discussions

The proposed system's performance is assessed using three measures: precision, recall, and the F-measure. Precision gauges the accuracy of the system and is calculated by dividing the number of relevant web pages retrieved by the total number of web pages retrieved. Recall measures the quantity of relevant web pages retrieved and is calculated by dividing the number of relevant web pages retrieved by the total number of relevant web pages retrieved by both the proposed system and Google. Average Precision (AP) or Average Relative Recall (AR) values are calculated as the average of all precision or relative recall values for single-word and multi-word queries, respectively. Mean Average Precision (MAP) or Mean Average Relative Recall (MAR) is determined by computing the mean of the average precision or average relative recall values for single-word and multi-word queries. The F-measure is a composite metric that balances precision and recall, yielding a single score that encapsulates the overall system performance. The evaluation criteria for the proposed system are described below.

$$\text{Precision} = \frac{\text{Total Web pages relevant for each query}}{\text{Total Web pages retrieved for that query}} \quad (13)$$

$$\text{Recall} = \frac{\text{Total Web pages retrieved by proposed system}}{\begin{array}{c}\text{Total Web pages retrieved by}\\ \text{the proposed system and Google}\end{array}} \quad (14)$$

$$\text{F} - \text{measure} = \frac{2 \times \text{ MAP} \times \text{ MAR}}{\text{MAP} + \text{MAR}} \quad (15)$$

The search results of the proposed system are compared to those of Google. Google's page ranks are checked by either installing the Google toolbar or using one of the page rank checking tools, such as www.prchecker.info. When compared to Google's search results, the proposed system consistently exhibits superior relative recall, precision, and F-measure values. This demonstrates that the proposed system performs better in terms of quantity, accuracy, and overall performance. The proposed system has been tested with single-word and multi-word queries. Figs. 2-7 present the results graphically.

The precision values of the proposed system, compared to Google, for single-word and multi-word queries are displayed in Figs. 2 and 3, respectively. These values assess the accuracy and relevance of the retrieved results, calculated as the precision in relation to the total number of results. In both figures, the Y-axis represents accuracy values, which indicate the ratio of relevant web pages retrieved to the total number of retrieved web pages for each query. The X-axis represents various queries, labeled as Q1 to Q10. Both figures clearly demonstrate that the proposed system outperforms Google in terms of precision for both multi-word and single-word queries. This suggests that the proposed system excels in locating relevant web pages across a wide range of search query types.

Figs. 4 and 5 focus on the relative recall values of the proposed system compared to Google for both multi-word and single-word queries. Recall is a measure of how effectively a system retrieves all relevant results, and relative recall indicates the ratio of relevant results retrieved by a system compared to the total number of relevant results available. In both figures, the X-axis represents different queries, labeled as Q1 to Q10, and the Y-axis represents relative recall values. Both figures clearly demonstrate that the proposed system outperforms Google in terms of relative recall for both multi-word and single-word queries. This suggests that the proposed system is more effective at retrieving a higher proportion of relevant web pages, making it advantageous for users who prioritize comprehensive and relevant search results.

TABLE XII
Ranked Web Pages Based on their Similarity to the User's Query Using the Proposed New Similarity

No.	URL	Semantic Similarity	Syntactic Similarity	Sentiment Similarity	Average Similarity
1	https://en.wikipedia.org/wiki/Information_retrieval	0.9961	0.9547	0.9871	0.9793
2	https://www.geeksforgeeks.org/issues-in-information-retrieval	0.9944	0.9172	0.9663	0.9593
3	https://nlp.stanford.edu/IR-book/information-retrieval.html	0.9914	0.8687	0.9601	0.9401
4	https://paperswithcode.com/methods/category/information-retrieval-methods	0.9897	0.8475	0.9856	0.9409
5	https://www.sciencedirect.com/topics/computer-science/information-retrieval-systems	0.9860	0.8034	0.9678	0.9191
6	https://www.sciencedirect.com/topics/computer-science/information-retrieval-systems	0.9834	0.7838	0.9854	0.9175
7	https://medium.com/@soumya.vkshukla/information-retrieval-a-brief-overview-173bba8fe0e9	0.9799	0.7605	0.9760	0.9055
8	https://www.linkedin.com/pulse/information-retrieval-basics-sagar-khatavkar	0.9769	0.7280	0.9648	0.8899
9	https://www.kaggle.com/code/vabatista/introduction-to-information-retrieval	0.9740	0.6866	0.9802	0.8803
10	https://en.wikipedia.org/wiki/Compound-term_processing	0.9727	0.6383	0.9695	0.8601

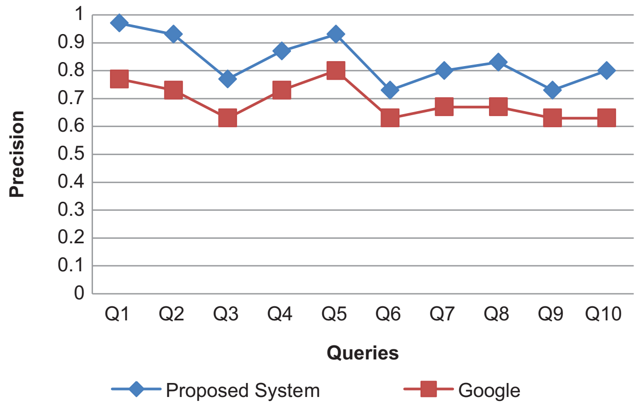

Fig. 2. Precision of single-word queries.

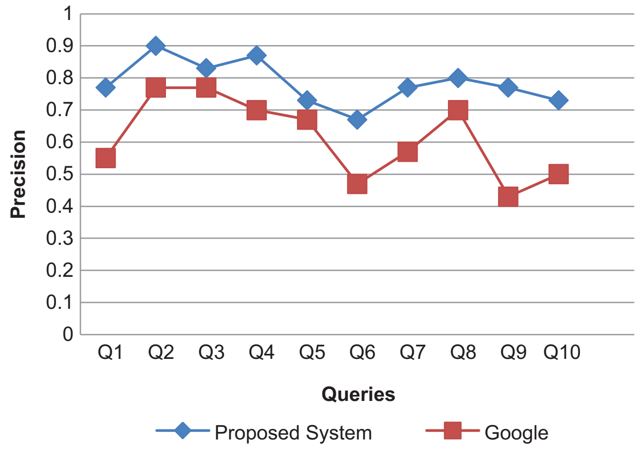

Fig. 3. Precision of multi-word queries.

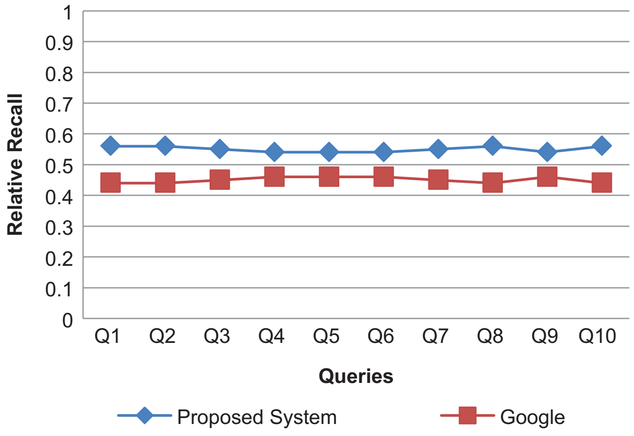

Fig. 4. Relative recall of single-word queries.

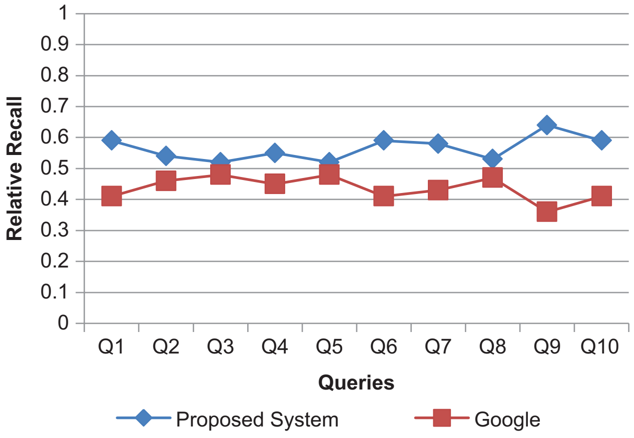

Fig. 5. Relative recall of multi-word queries.

Fig. 6 presents the AR and AP values of the proposed system and Google for both multi-word and single-word queries. The X-axis displays the evaluation measures, including APS (Average Precision of Single-Word Queries), APM (Average Precision of Multi-Word Queries), ARS (Average Relative Recall of Single-Word Queries), and ARM (Average Relative Recall of Multi-Word Queries), while the Y-axis represents the values of AR and AP. The graph clearly illustrates that, for both multi-word and single-word queries, the AR and AP values of the proposed system surpass those of Google. In other words, on average, the proposed system retrieves a higher proportion of relevant web pages compared to Google's search results for both query types. Users who

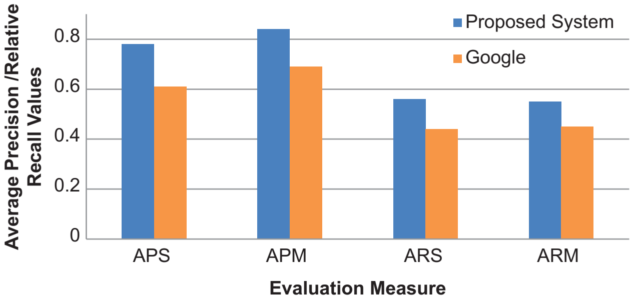

Fig. 6. Average Precision and Average Relative Recall of the proposed system and Google.

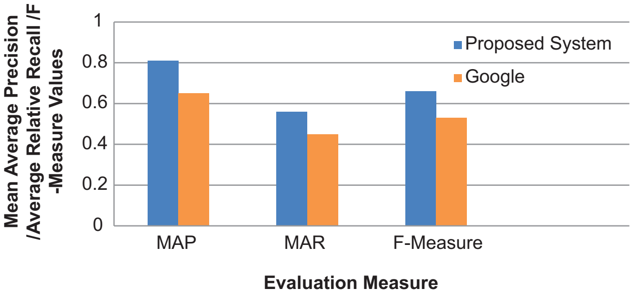

Fig. 7. F-measure of the proposed system and Google.

prioritize accuracy and comprehensiveness in their search results may find the proposed system to be the superior choice due to its consistently higher AP and AR performance compared to Google.

Fig. 7 compares the F-measure values of the proposed system and Google. The X-axis represents the evaluation measures: MAP, MAR, and F-measure. The Y-axis represents the values of these measures in fractions. Fig. 7 clearly demonstrates that the proposed system outperforms Google in terms of relative recall, precision, and F-measure. This indicates that the proposed system excels in retrieving relevant web pages with greater accuracy and comprehensiveness compared to Google's search results.

VI. Conclusions

This paper aims to develop an efficient approach for Web IR to enhance the search process and assist users in finding relevant content based on their queries. The proposed system ranks web pages by considering both the structural links of the pages, the content within them, and log files, resulting in high precision. By adopting this approach, we achieve high-quality results. The proposed method involves data collection from Google using an API, followed by storing this data in a database for further analysis. Utilizing an API for data retrieval offers advantages such as automated data collection, customizable requests, speed, efficiency, security, standardization, scalability, and real-time data updates. These qualities make it a preferred choice for many developers and researchers. Rate limiting is a challenge associated with using API for data collection from Web sites. Rate limiting refers to a restriction on the number of API requests that can be made within a certain time frame. This is done to prevent server overload and maintain Web site performance. Therefore, developers must optimize their code and implement techniques such as throttling to avoid exceeding the rate limit. To address the limitations of traditional ranking methods, such as Google's PageRank algorithm, a web page ranking method based on the number of links visited is proposed. This approach integrates semantic metadata analysis and considers factors such as visitation frequency, regional relevance, and related topics and queries to re-rank the links retrieved from Google. The goal is to provide more relevant, personalized, and context-aware search results. However, there are extra effort on crawlers to fetch the visit counts of Web pages from Web servers. Initiating the web crawling process with seed URLs derived from the top links of each semantic metadata criterion is a strategic approach that allows for targeted and efficient URL collection. Focusing the crawling effort on specific web pages, deemed relevant based on semantic criteria, enhances the quality and relevance of collected data, leading to improved retrieval and analysis tasks. Term-weighting schemes are vital in the VSM for document representation and IR. By judiciously selecting and combining term-weighting schemes, VSM can effectively rank documents, improving the accuracy and relevance of search results. Improved SVD is a powerful technique for dimensionality reduction in text mining and other data analysis tasks. It effectively filters out noise and less significant variations in data, resulting in a cleaner representation and enabling more efficient and effective data analysis with reduced memory and computational requirements. Finally, this paper introduces a new similarity measure for document retrieval, offering a more comprehensive understanding of how documents relate to user queries. This measure enables a more precise ranking of content.

VII. Future Work

Research work illustrated in this paper can be stretched in many directions that will help in enhancing the results thus obtained.

1. Web page data may be collected by using screen scraping, where data are extracted from the source code of a Web site with an HTML parser or regular expression matching.
2. The weight of a term in a document vector may be determined by using information gain.
3. The high dimensionality problem may be addressed by performing the rough set based on feature selection and by designing the rough set based on membership functions.
4. The system may be modified to have the provision of refining the input query by using the relevance feedback technique.
5. Evaluation can be carried out on techniques for documents in various languages, as well as on the study of how language affects the performance of the retrieval process.

References

Afolabi, I.T., Makinde, O.S., and Oladipupo, O.O., 2019. Semantic web mining for content-based online shopping recommender systems. *International Journal of Intelligent Information Technologies*, 15(4), pp.41-56.

Al-Anzi, F., and Abuzeina, D., 2020. Enhanced latent semantic indexing using cosine similarity measures for medical application. *International Arab Journal of Information Technology*, 17(5), pp.742-749.

Alhaidari, F., Alwarthan, S., and Alamoudi, A., 2020. User preference based weighted page ranking algorithm. In: *ICCAIS 2020-3rd International Conference on Computer Applications and Information Security*, pp.1-6.

Ali, F., and Khusro, S., 2021. Content and link-structure perspective of ranking webpages: A review. *Computer Science Review*, 40, p.100397.

Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E.D., Gutierrez, J.B., and Kochut, K., 2017. A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques. *Journal of Intelligent Information Systems,* 2017, 1(1), pp.1-13

Ghani, W.A., and Hussain, A., 2021. Applying similarity measures to improve query expansion. *Iraqi Journal of Science*, 62(6), pp.2053-2063.

Guwta, M., 2021. *Information Retrieval for Silt'e Text Using Latent Semantic Indexing*. M.C. Thesis. Bahir Dar University.

Hazarika, D., Konwar, D., and Bora, D.J., 2020. Sentiment Analysis on Twitter by Using TextBlob for Natural Language Processing. In: *Proceedings of the International Conference on Research in Management and Technovation 2020*. Vol. 24, pp.63-67.

Jain, S., Jain, S.C., and Vishwakarma, S.K., 2020. Analysis of text classification with various term weighting schemes in vector space model. *International Journal of Innovative Technology and Exploring Engineering*, 9(10), pp.390-393.

Jain, S., Vishwakarma, S., and Jain, S.C., 2023. Analysis of term weighting schemes in vector space model for text classification. *Journal of Integrated Science and Technology*, 11(2), p.469.

Joby, P.P., 2020. Expedient information retrieval system for web pages using the natural language modelling. *Journal of Artificial Intelligence and Capsule Network*s, 2(2), pp.100-110.

Kleinberg, J.M., 2011. Authoritative sources in a hyperlinked environment. In: *The Structure and Dynamics of Networks*. Princeton University Press, Princeton, pp.514-542.

Lu, J., Henchion, M., and Namee, B.M., 2020. Diverging Divergences: Examining Variants of Jensen Shannon Divergence for Corpus Comparison Tasks. In: *LREC 2020-12th International Conference on Language Resources and Evaluation, Conference Proceedings*. Vol. 2, pp.6740-6744.

Mustafa, A.B., Ghulam, S.K., Naadiya, M., and Sheeba, M., 2022. Web content mining techniques for structured data: A review. *Sindh Journal of Headways in Software Engineering*, 1(1), pp.1-10.

Nassar, M.O., Kanaan, G., and Awad, H.A.H., 2010. Comparison between Different Global Weighting Schemes. In: *Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010*. Vol. I, pp.690-692.

Patel, S.H., and Desai, A.A., 2019. Link analysis to discover relevant documents using information retrieval. *International Journal of Computer Application*s, 178(10), pp.23-27.

Payal, L.S., 2020. A study of different web mining types. *Anveshana's International Journal of Research in Engineering and Applied Sciences*, 5(3), pp.30-33.

Phyu, A.P., and Thu, E.E., 2021. Short survey of data mining and web mining using cloud computing. *International Journal of Advanced Networking and Applications*, 12(05), pp.4725-4731.

Qi, Q., Hessen, D.J., and van der Heijden, P.G.M., 2023. Improving Information Retrieval Through Correspondence Analysis Instead of Latent Semantic Analysis. *Journal of Intelligent Information Systems*, 2023, 1(1), pp.1-44.

Rathi, R.N., and Mustafi, A., 2023. The importance of term weighting in semantic understanding of text: A review of techniques. *Multimedia Tools and Applications*, 82(7), pp.9761-9783.

Reddy, K.P., Reddy, T.R., Naidu, G.A., and Vardhan, B.V., 2018. Impact of similarity measures in information retrieval. *International Journal of Computational Engineering Research*, 8(6), pp.54-59.

Robert, B., and Brown, E.B., 2004. *The PageRank Citation Ranking: Bringing Order to the Web*. Vol. 1, University of Pennsylvania, Philadelphia, PA, pp.1-14.

Shahmirzadi, O., Lugowski, A., and Younge, K., 2019. Text Similarity in Vector Space Models: A Comparative Study. In: *Proceeding-18th IEEE International Conference on Machine Learning and Applications, ICMLA 2019*, pp.659-666.

Sharma, D., Shukla, R., Giri, A.K., and Kumar, S., 2019. A Brief Review on Search ENGINE Optimization. In: *Proceedings of the 9th International Conference On Cloud Computing, Data Science and Engineering, Confluence 2019*, pp.687-692.

Sharma, P.S., Yadav, D., and Garg, P., 2020. A systematic review on page ranking algorithms. *International Journal of Information Technology*, 12(2), pp.329-337.

Sharma, P.S., Yadav, D., and Thakur, R.N., 2022. Web page ranking using web mining techniques: A comprehensive survey. *Mobile Information Systems*, 2022, p.7519573.

Ilo, P.I., Nkiko, C., Izuagbe, R., and Furfuri, I.M.M., 2023. *Course Guide Lis 303 Information Retrieval (Cataloguing ii)*. National Open University of Nigeria, Nsukka.

Thakur, N., Mehrotra, D., Bansal A., and Bala M., 2019. Comparative analysis of ranking functions for retrieving information from medical repository. *Malaysian Journal of Computer Science*, 32(1), pp.18-30.

Tyagi, N., and Gupta, S.K., 2018. Web structure mining algorithms: A survey. *Advances in Intelligent Systems and Computing*, 654, pp.305-317.

Wang, J., and Dong, Y., 2020. Measurement of text similarity: A survey. *Information*, 11(9), p.421.

Wu, H., and Gu, X., 2014. Reducing Over-weighting in Supervised Term Weighting for Sentiment Analysis. In: *COLING 2014-25th International Conference on Computational Linguistics, Proceedings of COLING 2014: Technical Papers*, pp.1322-1330.

Xing, W., and Ghorbani, A., 2004. Weighted PageRank Algorithm. In: *Proceedings-Second Annual Conference on Communication Networks and Services Research*, pp.305-314.

Zheng, W., and Fang, H., 2010. *A Retrieval System based on Sentiment Analysis*. HCIR. [Preprint].