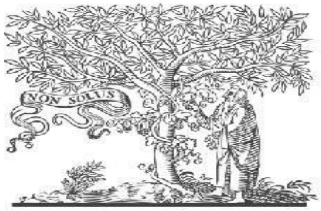
International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

COPY RIGHT



ELSEVIER SSRN

IJIEMR Transactions, online available on 10th Apr 2023. Link :http://www.ijiemr.org/downloads.php?vol=Volume-12&issue=Issue 04

10.48047/IJIEMR/V12/ISSUE 04/111

Title **BUILDING SEARCH ENGINE USING MACHINE LEARNING TECHNIQUE**

Volume 12, ISSUE 04, Pages: 884-890

Paper Authors

Mr.B.Kalyan Chakravarthy, A.Venkata Vyshnavi, CH.Manasa, J.Aswitha, B.Sravani

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

BUILDING SEARCH ENGINE USING MACHINE LEARNING TECHNIQUE

Mr.B.Kalyan Chakravarthy, Assistant Professor, Department of IT,
Vasireddy Venkatadri Institute of Technology, Nambur, Guntur Dt., Andhra Pradesh.

A.Venkata Vyshnavi, CH.Manasa, J.Aswitha, B.Sravani
UG Students, Department of IT,
Vasireddy Venkatadri Institute of Technology, Nambur, Guntur Dt., Andhra Pradesh.
kalyanchakravarthy.battula@gmail.com

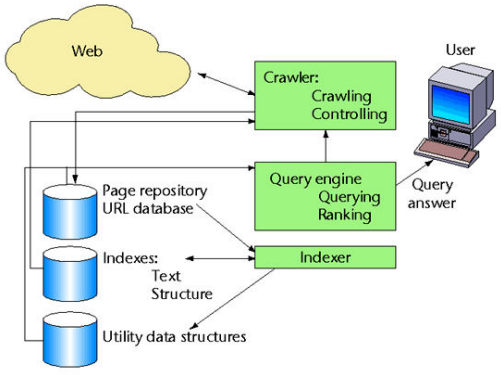
Abstract

The proliferation of information available on the internet has led to an exponential increase in the number of web pages, making it difficult for users to find relevant information efficiently. Search engines have become the primary means of accessing information on the internet. However, current search engines are not personalized, and search results are often influenced by a user's search history, location, and other factors. This project aims to develop a personalized search engine using machine learning techniques. The search engine will use data from a user's search history, browsing history, and social media activity to build a personalized profile for the user. The system will then use this profile to provide search results that are tailored to the user's interests and preferences. The search engine will be built using Python and will leverage machine learning libraries such as TensorFlow, Keras, and Scikit-Learn. The system will be trained using a dataset of web pages and user activity data. The system will be evaluated using metrics such as precision, recall, and F1 score.

Keywords: Query parsing, Search Engine, Crawler, Indexing, and Machine Learning.

Introduction

The Internet is composed of multiple systems linked together. Each website contains a vast amount of information within its pages. When a user wants to find something, they enter a keyword, which is a set of words taken from their search query. Sometimes the user may input incorrect syntax. Search engines are crucial in this situation because they provide a simple method of searching for user queries and show relevant web links as results.



1) Web crawler

Web crawlers are used to collect data on a website and the links that are connected to it. Their only goal is to compile and save information about the World Wide Web in a database.

2) Indexer

An indexer is a crucial component of a search engine that is responsible for analyzing and organizing the content of web pages in a way that enables efficient and accurate retrieval of information in response to user queries. It works by processing and storing information from crawled web pages, such as the words, phrases, and metadata, and creating an index that maps these items to their corresponding web pages. This index serves as a catalog that allows the search engine to quickly find and retrieve relevant results for a given search query. The indexer is a complex system that involves a combination of techniques, such as natural language processing, machine learning, and data mining, to accurately and efficiently index the vast amount of information available on the web.

3) Query engine

The Query Engine's main job is to respond to user inquiries with pertinent information based on their keywords.

The PageRank algorithm uses a variety of techniques to rank URLs within the search engine to achieve this.

This study uses machine learning methods to get the best website address for a given keyword, with the PageRank algorithm's output serving as the machine learning algorithm's input.

2. Literature Survey

Building a search engine requires a comprehensive literature survey that covers various areas of study such as information retrieval, natural language processing, machine learning, and web mining. Below are some papers that could be useful for building a search engine:

1. "A Survey of Information Retrieval and Extraction Technologies" by T. Sebastian, S. H. Myaeng, and W. B. Croft. This paper presents a comprehensive overview of information retrieval techniques, including document and query processing, ranking algorithms, and evaluation metrics.
2. "Natural Language Processing with Python" by Steven Bird, Ewan Klein, and Edward Loper. This book covers the fundamental concepts of natural language processing (NLP) and provides practical examples using Python.
3. "Building a Web Search Engine: An Information Retrieval Perspective" by Ricardo Baeza-Yates and Berthier Ribeiro-Neto. This book provides a comprehensive overview of the

algorithms and techniques used in modern search engines.

4. "A Comparative Study of Stemming Algorithms" by Jivani and Kotecha. This paper compares various stemming algorithms, which are used to reduce words to their roots, in the context of information retrieval.

3. Problem Statement

1) How do I start building a search engine?

2) How do I get related URLs to the top of the search results?

Our goal is to build a search engine that makes use of machine learning to improve the accuracy of search results. This system's main objective is to place the website that best matches user queries at the front of the search results.

4. Methodology

Our goal is to develop a search engine that uses machine learning to increase its precision in comparison to other search engines that are already on the market.

The following steps are involved in creating the search engine:

1) Using a web crawler to get information from the Internet.

2) Using tools for natural language processing to carry out data cleaning.

3) Exploring and comparing available page ranking algorithms.

4) Integrating the chosen algorithm with contemporary machine learning technologies.

5)creating a search engine to deliver efficient search outcomes for user queries.

A Use a web crawler to gather information from the WWW.

During this phase, we employed a web crawler that relies on keywords to obtain information from the internet. The process begins with a seed URL, which is used to access the website page. Once on the page, the crawler searches for and retrieves all hyperlinks present on the page, saving them in a queue for later processing. This ensures that data is accurately collected from all web pages. The crawler then filters out URLs that are relevant to specific keywords.

The algorithm follows these steps:

Step 1:Start with the initial URL

Step 2:Initialize a queue (q)

Step 3:Extract URLs from the queue (q).

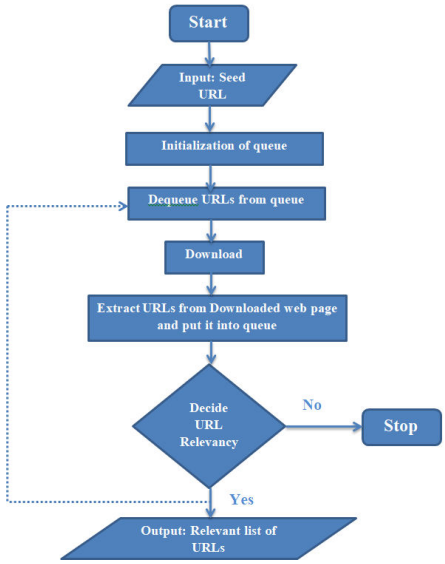
Step 4: Download the URL-linked web pages.

Step 5: Take note of every URL on the web pages you downloaded.

Step 6:Put the extracted URLs into the queue (q).

Step 7: Repeat step 1 till you get more pertinent results.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org



Flowchart for a keyword-focused web crawler in Figure 2 [2]

B. Use NLP to perform data cleaning

Once data is collected from the World Wide Web using a web crawler, the next step involves data cleaning. This process employs NLP techniques to preprocess the data, ensuring that any extraneous data is removed.

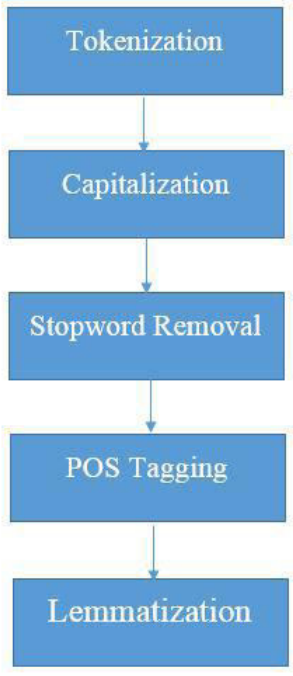


Fig. 3. NLP steps for data cleaning

The detailed process for data cleansing with NLP is shown in Figure 3.

1) Tokenization = Tokenization is the process of separating phrases or whole phrases from web page sequences.

2) Upper case = The most typical strategy is to simplify everything on a web page by converting it to lower case.

3) Eliminating stop words = Instead of presenting crucial information, web pages frequently utilise words to join sentences together. Here, such words must be eliminated.

4) Parts of Speech Tagging: This method, which also applies to user requests in search engines, breaks the sentence up into tokens and assigns meanings to each token.

5) Lemmatization: By eliminating extraneous characters, lemmatization reduces words to their root..

C. Examine and contrast the current page ranking formula.

TABLE I

COMPARISON BETWEEN PR, WPR, AND HITS [7]

Criteria	PageRank (PR)	Weighted PageRank (WPR)	HITS
Working	This algorithm calculates the page score at the time the pages are indexed.	Web page weight is calculated based on inbound and outbound links of importance web page.	It calculates hub and authority score for each web page.
Input Parameter	Incoming links	Incoming and outgoing links	Content, incoming and outgoing links
Algorithm Complex-ity	O(log N)	< O(log N)	< O(log N)
Quality of Results	Good	More than PageRank	Less than PageRank
Efficiency	Medium	High	Low

The system is best suited to the weighted PageRank algorithm since it offers greater

accuracy and efficiency than other algorithms (see Table 1).

D. incorporate cutting-edge machine learning methods with the selected page rank algorithm

The best PageRank algorithm is chosen and implemented, and the top result is then fed into a machine learning algorithm to discover the most pertinent web page based on user searches. Web features are grouped into three categories to do this:

1) Page content
2) Content on pages that are nearby
3) Link analysis

These characteristics are regarded as input characteristics for ANN, SVM, and Xgboost. The chosen PageRank algorithm is then combined with the feature that yields the best accurate results to identify the URL of the pertinent web page for user searches**.**

E. Construct a query engine to show the user's query's effective results.

The query engine is then created to process user queries and deliver pertinent search results. The output of the machine learning algorithm will be used to construct the search results, showing the user the web addresses of the most pertinent pages.

5. Implementation

Here is a possible rephrased version:We have implemented several algorithms, including

1. Support vector machine
2. Artificial Neural Network
3. XGBoost

The most accurate algorithm, as determined through experimentation, is used in conjunction with the PageRank algorithm.

5.1 Support Vector Machine

An SVM was incorporated as it showed superior performance compared to other algorithms. To deal with the non-linear separability of the dataset, a nonlinear SVM was used, with options for different types of kernels such as Rbf, poly, and sigmoid. The input features for the SVM model were the 14 selected features. Based on these qualities, the SVM forecasted whether each web page in the test set was pertinent to the given question. The model's effectiveness was assessed in light of the outcomes..

5.2 Artificial Neural Network

The neural network architecture comprises an input layer, a hidden layer, and an output layer. Each input layer node represents one of the 14 feature values of a web page. The output layer has a single node that determines a web page's relevance. A grid search was used to determine the number of nodes in the hidden layer, which was set to 7. The process was repeated 150 times with a stack size of 10, and the results were saved for performance evaluation.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

5.3 XGBoost

The system uses XGBoost , which is an ensemble learning method based on boosting. It improves accuracy and speed by using gradient-boosted decision trees. The 14 input features are equally considered, and gbtree -based boosters are utilized. Through parameter tuning and cross-validation, the maximum depth size is set to 4, and the number of classifiers is set to 50. On the basis of preliminary experiments, these parameters were chosen.

A. Performance

1) Precision

The following formula is used to determine accuracy.

accuracy = (number of documents correctly classified)/ (total number of documents)

The accuracy of each algorithm is displayed in the following table. Among them all, XGBoost has the best accuracy.

TABLE II

ACCURACY OF DIFFERENT ALGORITHM

No.	Algorithm	Accuracy
1	SVM	89.50
2	ANN	91.35
3	XGBoost	92.59

6. Results

After the user successfully logs in, he can check the page rank of the desired file. The following page appears

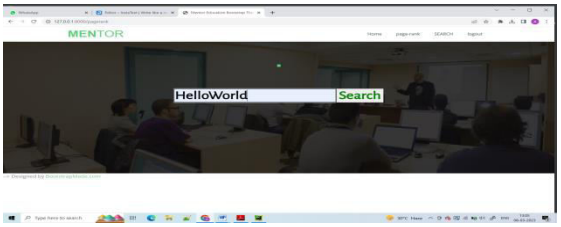


Fig. 6.1.1 Search for page rank

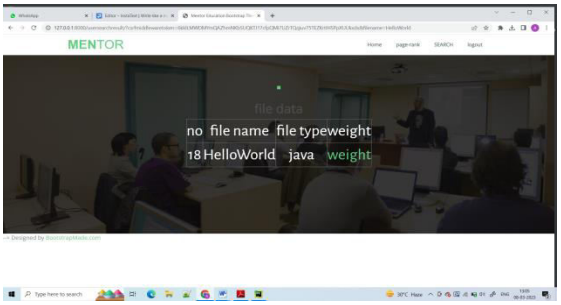


Fig 6.1.2 Succesful Retrieval of File

Then, if we click on weight, it shows the weight.

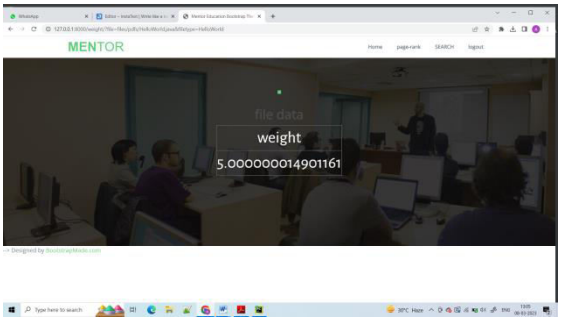


Fig. 6.1.3 Page displaying weight

After getting the weight, the user can search for the optimal file size.

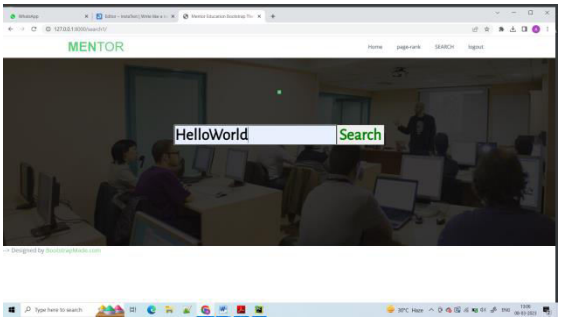


Fig. 6.1.4 Search Page

If the file is available, he gets a list of available sizes.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

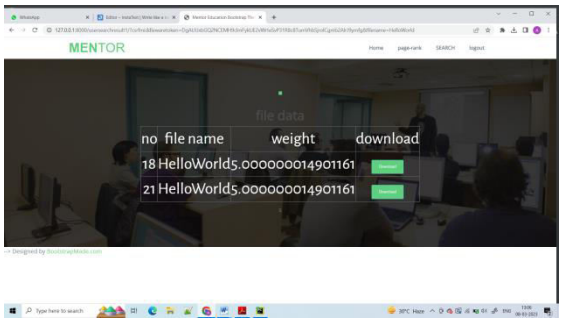


Fig 6.1.5 Page with all sizes

7. Conclusion

Using a search engine to find information can save time by displaying relevant URLs for a given keyword. Accuracy is crucial in achieving this goal. Based on the observations made, XGBoost outperforms SVM and ANN, making it a better choice. Thus, it is anticipated that a search engine created combining XGBoost plus the PageRank algorithm will have improved accuracy..

8. Future Work

The focus of our work in this article was to create a search engine that retrieves files. However, our future plans involve expanding its functionality to include searching for files that contain specific keywords.

9. References

Here are some references that can be useful in building a search engine project:
[1] "Introduction to Information Retrieval" by Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze.
[2] "Search Engines: Information Retrieval in Practice" by Bruce Croft, Donald Metzler, and Trevor Strohman.
[3 "Learning to Rank for Information Retrieval" by Tie-Yan Liu.
[4 "Modern Information Retrieval" by Ricardo Baeza-Yates and Berthier Ribeiro-Neto.
[5] "Web Information Retrieval" by Stefano Ceri, Alessandro Bozzon, Marco Brambilla, and Emanuele Della Valle.
These books cover a wide range of topics related to building search engines, including information retrieval, machine learning, natural language processing, and web crawling. They can provide a solid foundation for anyone looking to develop a search engine project.