2024 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS)
Pune, India. Oct 17 - 19, 2024

Detecting plagiarism using Latent Semantic Analysis and Cosine Similarity Approach

Vedant Wagh
Government College of Engineering,
Aurangabad (Chh. Sambhajinagar)
waghvedant21@gmail.com

Dr. Shilpa Laddha
Assistant Professor,
Government College of Engineering,
Aurangabad (Chh. Sambhajinagar)
kabrageca@gmail.com

Prashant Kadam
Government College of Engineering,
Aurangabad (Chh. Sambhajinagar)
kadamprashant9822@gmail.com

2024 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS) | 979-8-3503-5434-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICBDS61829.2024.10837475

***Abstract*--- The act of using someone else's words and/or ideas and passing them off as your own is known as plagiarism. The act of presenting one's own thoughts or works as those of another person. Plagiarism, that point, is the willful taking of another person's thoughts, words, or works. Finding instances of plagiarism and/or copyright violations in a work or document is what a plagiarism tool achieves. We are developing an Offline Plagiarism Tool to avoid plagiarism. This research employs latent semantic analysis and cosine similarity to compare the papers' similarity. must steer clear of any accidental or purposeful plagiarism. To arrange tasks by employing these tools to obtain the similarity score to ensure that the text is truly approved by the organizations and to prevent errors in paraphrase. Two strategies have been built to determine the similarity score of documents based on word matching and semantic meaning.**

This tool additionally shows to the user the percentage of plagiarism in their document and the necessity for changes in an eye-catching graphical format. This procedure will undoubtedly assist in determining whether or not content has been duplicated, and user-generated datasets have been gathered for this purpose. It aids in improving the ability to organize and paraphrase text as well as to show different graphs so that the user can quickly comprehend the outcomes.

Keywords: Plagiarism Detector, text similarity, Cosine similarity, Semantic Analysis, Language processing, Text preprocessing, Words Matching, Cosine vector calculation, Text Analysis, Singular Value decomposition, Semantic matching, Meaning Detection

I. INTRODUCTION

Within the academic community, plagiarism by students is typically seen as a highly serious infraction that carries severe penalties, including failing grades for the offending work, the entire course, or even expulsion from the university. Suspension or expulsion may be the result of a student's frequent plagiarism or serious plagiarism offenses (such as buying an assignment). However, to impose sanctions and considering the originality as a parameter, plagiarism needs to be detected. In order to overcome this problem, we have addressed the plagiarism tool with the help of that students previously knows about the similarity score and can organized the data.

Plagiarism describes our research and conclusions and serves as a means of demonstrating that our work is entirely original and unreproduced from outside sources.

It is too much hazardous data theft is a possibility if our paper has excessive amounts of confidential information. The process is also online, so if the document gets caught up in the workflow, data can be lost. Online tools are essential if we want to review our documents in particular regions exclusively, such as for departmental purposes, as this ensures privacy and eliminates the possibility of data theft.

II. OBJECTIVE

Our Offline Plagiarism Tool's primary objectives are –

- **To apply the current algorithm in accordance with the specific requirements of our system.** While predefined methods such as cosine similarity and latent semantic analysis are easily accessible, we attempt to create our own modules that are tailored to the system and document we use. For example, we reduce the document's dimension, cut down on unnecessary calculations, and restrict their use to internal use only.in order for document comparison to take place inside the dataset and data security.
- **To obtain precise outcomes. Score takes into account more than just the words, including paraphrasing and well-meaning.** Preprocessing our content is a crucial step before submitting it for plagiarism detection, ensuring that comparisons are made only on relevant terms. Following processing, when the value is computed, the actual scores show the phrase, and additional computation is based on those scores.
- **To offer a very user-friendly interface that is even suitable for non-techies to utilize.** It is not required of the user to be fully conversant with document preprocessing and document checking. The principle of data abstraction is to present users with only the most crucial elements while suppressing irrelevant ones. in order for even the average individual to review documents, given that the interface is designed to be user-friendly and accessible.

III. LITERATUR SURVEY

In [1], the author proposed a user interface that would allow users to make comparisons and take documents as inputs. It also offers one of the most effective processing of documents flowcharts that can be found.

The author in [2] proposes a few natural language processing techniques, such as corpus-based, knowledgebased, and string-based approaches, for the semantic analysis of the document. Semantic meaning is the primary basis for classification of texts.

A text summarizing technique to extract only relevant material is defined in this research [3]. The document under comparison contains enormous amounts of data, however first we must preprocess the data. It offers a few text extraction techniques to ensure that only pertinent and significant content is compared.

An intelligent method for identifying semantic plagiarism in scientific publications is suggested in this study [4]. A corpus containing the text of original scientific articles has been developed in order to compare suspicious documents with it and identify instances of plagiarism. Using the Mini-Batch K-Means clustering algorithm, the documents are grouped into many groups and then placed into a designated category.

The technique to cluster words from sentences and group comparable ones is covered in this study [5]. The three most popular approaches for identifying phrase similarity are vector-based, word-to-word, and structure based.

In order to determine similarity, this paper [6] identified certain drawbacks with text clustering and suggested an alternative approach. These restrictions will be addressed by utilizing XLNet in conjunction with a DKMclustered Bi-LSTM model. The generalized autoregressive model, which creates the highlighted vectors from preprocessed data, is the main contribution of the presented study.

There are certain limitations in references that we have attempted to overcome. For example, in [2], the author suggests additional methods such as knowledge-based and string-based for text classification in order to get a clearer result. However, by using a single word-based method, the same result is obtained and the computation time for words is decreased.

A few of the publications we looked at addressed text preparation techniques [2][3], suggesting strategies to remove just pertinent data so that our algorithm processes less input in less time.

[4] Author of the document utilized the K-means clustering technique to group the documents because it was a little complex to analyze the results for the complete text. By using the document as a guide, we attempted to group the terms that were related so that analysis of documents that are compared easily are those that use word-based indexing.

The author in [7] defined word-to-word embedding mapping for documents, noting that word embedding is useful for offline purposes but becomes more dangerous when it occurs in an online tool.

Similarly, word embedding is used in [7] to create the cosine similarity algorithm for word-to-word mapping detection.

Paper [8] provides a definition. Latent semantic analysis (LSA) is a technique that uses certain mathematical calculations to analyze text and examine the relationships between terms and documents within a corpus. Singular value decomposition is used to break down the corpus of related words into manageable numerical representation. LSA demonstrates how a single value decomposition may recognize terms with numerous meanings, collapse several terms with the same semantic, and represent texts in a conceptual space which is in lower dimensional.

This study [9] examines Along with the traditional text processing procedure in semantic analysis, the most popular models and techniques for semantic text processing are taken into account.

As was previously mentioned, there are some pretrained methods available that make certain calculations fairly complex because they rely on gathering words from the internet. We are creating this for internal use, and since the amount of data is small, there is no requirement to Apply that model. instead of adjusting the model's parameter, which is necessary for the computation to be done using our dataset.

IV. PROPOSED SYSTEM

We will be concentrating on comparing the input file with the files kept in our database as we construct an offline utility. Our project's main goal is to modify an existing algorithm to meet our specifications. The two primary categories of plagiarism detection tools are Cross lingual and Monolingual. We are working on our monolingual plagiarism detection technology. The primary goal of this project is to prevent the uploading of duplicate data that already exists. We are using the Python platform to design this system.

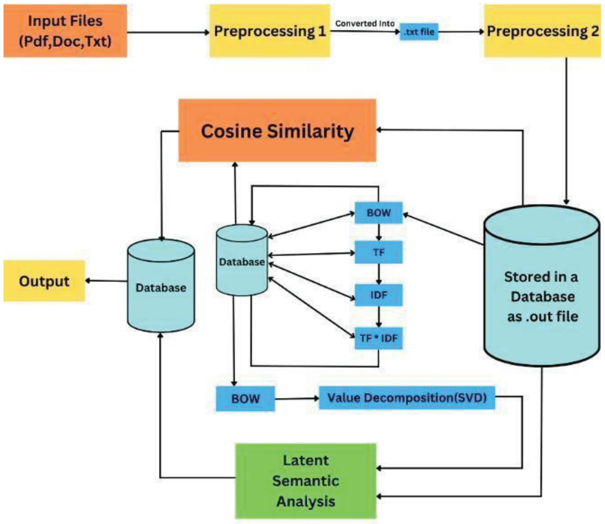


Fig. 1. Proposed System

Only files with data to be processed in the extensions.txt,.pdf, and.docx or.doc are accepted for upload by users. We must preprocess the data in order to compare only what is relevant while determining the similarity.

A. Preprocessing 1

Preprocessing is a method that includes converting unprocessed data into a comprehensible format. Preprocessing speeds up the similarity computation and facilitates document extraction by focusing just on pertinent content rather than the entire file's contents [9]. It essentially goes through various preprocessing techniques. such as

1) Tokenization

Text strings are divided into smaller units called "tokens" by the process of tokenization. Sentences can be tokenized into words, and paragraphs into sentences.

For instance, "Never give up."

Output: "Never," "give," and "up"

2) *Eliminating of stop words and punctuation*

Stop words are frequently used terms. These terms and punctuation do not truly indicate any significance because they are not useful when comparing two writings. Therefore, in order to reduce computing time and effort while processing vast amounts of text, we can delete stop words and punctuation.

3) *Lemmatization*

The process of returning a word to its simplest form is called lemmatization. It correctly reduces the inflected words, guaranteeing that the original word is a part of the language. For instance: Caring -> Care

4) *Lower casing*

Lowercasing refers to Lower case conversion of a word (NLP -> nlp). Even if the words "Book" and "book" have the same meaning, they are displayed as two distinct words when they are not in lower case.

Since working with.txt files are generally fairly simple, we will convert all preprocess files to.txt.

B. *BOW (Bag of Words)*

An approach used in NLP and information retrieval is the bag-of-words (BOW) module. BOW representation consists of two elements:

1) a list of well-known words, 2) a tally of the terms that are known.

One of the most crucial modules is the bow module, which is utilized for both CS and LSA. Preprocessing will be place when a user uploads a new document. After all text has been divided into words, the bow module will run. Initially, every word will be kept in a single globally stated list, after which the word count in the document will be determined.

All words will finally be entered into a database along with their corresponding counts.

C. *TF (Term Frequency)*

The term frequency (TF) counts the instances of a word in a document.

TF (word) = (Count of each word in a document) / (Total number of words in the document) is the formula used to calculate TF.

For instance, in a 100-word paper, the word software appears six times.

Output: 6/100 = 0.06 for term frequency (software).

Preprocessing, the bow module, and the TF module are all executed when a user uploads a new document. The BOW table will first be searched for all words and their counts. The total number of words will then be determined, and the TF formula will be assessed. Eventually, a database including all words and their corresponding TF value will be created.

1) *IDF (Inverse document frequency)*

Every word's significance within the collection of documents is gauged by its inverse document frequency (IDF).

IDF calculation formula: log (total amount of documents / number of words with documents)

As an illustration, suppose there are 100 documents and the phrase "DEFEND" appears in 10 of them. IDF(DEFEND)= log (100/10) =1 is the output.

All previous modules are executed when a user uploads a new document, and as IDF depends on the quantity of documents for each value, all values must be updated. The previous table is removed whenever the IDF module is activated, and the logic for getting the quantity of documents from BOW and the frequency of a word is then carried out. All of the values are then updated and replaced with the appropriate formulas.

2) *TF-IDF Multiplier*

Text-based statistical weighting methods, or TFIDF, are employed in information retrieval. It is a technique for determining a term's significance in relation to a document or group of documents. To find the TF-IDF Multiplier, use this formula:

TF-IDF Multiplier(word) = TF (word) * IDF (word)

TF and IDF modules are required for the TF-IDF Multiplier to function. Following the execution of these two modules, the TF-IDF Multiplier will run. Initially, all word TF values will be obtained from the TF table, and all word IDF values will be obtained from the IDF table. After that, the formula is assessed. Eventually, a database including all words and their corresponding TFIDF value will be created.

3) *Cosine Similarity*

Regardless of the size of the documents, cosine similarity is a statistic that is used to assess how similar they are. The similarity between two vectors in an inner product space is measured by cosine similarity. It is a similarity rate that is calculated by multiplying the cosine angles of the two vectors under comparison. When the cosine similarity between the two vectors is 1, it indicates that the two vectors are similar because their cosine 0 degree is 1 and their cosine angle is less than 1.

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

In this formula, a and b are two vectors.

In numerator, 1st element of vector A will be multiplied with 1st element of vector B then 2nd element of vector A will be multiplied with 2nd element of vector B and this goes on till the last element and then their multiplication results will be added.

In denominator, now in the Denominator all the elements of vector A and vector B will be squared and added with the respective vector elements and then their square root will be calculated of the result obtained from 2 vectors and then the results are multiplied. At last value of numerator will be divided by value of denominator and thus cosine similarity of documents is obtained

D. *LSA (Latent Semantic Analysis)*

Latent Semantic Analysis (LSA) is a Natural Language Processing (NLP) approach. LSA's primary objective is to provide vector-based text presentations that produce semantic content. LSA calculates the textual similarity between two or more documents using vector representation. Previously known as Latent Semantic Indexing (LSI), LSA was enhanced for information retrieval. Finding a few documents that are similar to the query supplied among several document.

LSA should have a wide range of features, including vector representation based on word occurrences in documents, weighted key word matching, and key word matching. Additionally, the data is rearranged using singular value decomposition (SVD) in Latent Semantic Analysis (LSA). The SVD approach computes and reconfigures all of vector space's diminutions using matrices. Furthermore, the damnations the most significant assumption in LSA will be utilized to fine-tune the text's meaning; less significant assumptions will be disregarded. The values in vector space will be computed and arranged from most is the least important.

If the wards have a similar vector, a high rate of similarity search will be conducted for words. In order to outline the most important processes in LSA, first gather a sizable collection of pertinent text and then separate it into papers. Create a co-occurrence matrix for terms and documents in the second place, including the cell names document x, term y, and m for the dimensional value of the term, and n for the dimensional vector of the document. Third, each cell will be measured and its final values computed. To calculate all the diminutions and create three matrices, SVD will perform a wild ride.

The BOW module is required by the LSA module. To determine their semantic meaning and compare two or more papers, all words and their corresponding counts will be collected from the BOW table.

E. Database

SQLITE 3 is the database that we are using. The reason behind selecting Sqlite3is Installing an enterprise version is not necessary for every app or project we create.

Lastly, the system will show the similarity score of each user, indicating the percentage of their document that is similar to other documents that are available in a graphical format.

V. IMPLEMENTATION

Our system is implemented using Sqlite3 as the database and Python. There are two phases to implementation:

Before deadline: Determining the deadline is crucial because we must eventually stop accepting documents in order to conduct additional calculations. When the deadline is reached, the module stops accepting new documents from users, indicating that the dataset is now being processed in its entirety until the user has the authority to create new documents. create an account and send the file to be compared.

After deadline: When the deadline approaches, internal system modules and preprocessing begin running one by one, calculating BOW and cosine similarity, and finally, after running the semantic analysis module, the result is prepared for display. We may need some time to calculate on every document in the dataset that is currently available. and following all of the computation, when a user logs in using their created credentials, a list of all the documents that are available in the dataset is displayed. The user can select a document from the list, and the comparison result is displayed in an attractive graphical format as shown in the figure.

VI. FLOW OF SYSTEM

Users and administrators can use their special login credentials to access the system before the deadline passes. The user can upload a file to be compared, and once the deadline has passed, they can wait to see the similarity score. The administrator has access to all user data and file material that has been posted by users. Once the deadline has passed, cosine similarity and latent semantic analysis are calculated.

At this point, users are able to sign in and view their similarity score in relation to every document in the dataset as well as to individual documents in an accurate graphical format. The similarity score (cosine similarity and latent semantic analysis) of every document supplied by users prior to the deadline will be visible to the admin once all similarity calculations have been completed. flow chart for the system.

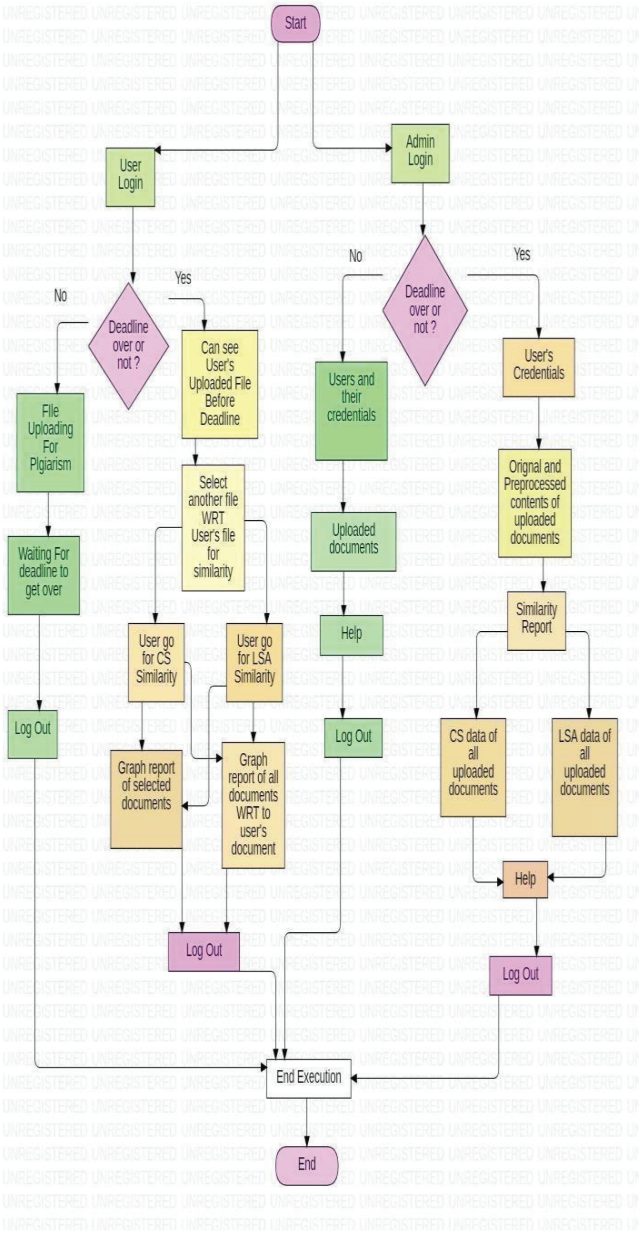

Start
User Login
Admin Login
No
Yes
Deadline over or not ?
No
Yes
Deadline over or not ?
File Uploading For Plgiarism
Can see User's Uploaded File Before Deadline
Users and their credentials
User's Credentials
Select another file WRT User's file for similarity
Orignal and Preprocessed contents of uploaded documents
Waiting For deadline to get over
Uploaded documents
Similarity Report
User go for CS Similarity
User go for LSA Similarity
Help
Log Out
Log Out
CS data of all uploaded documents
LSA data of all uploaded documents
Graph report of selected documents
Graph report of all documents WRT to user's document
Help
Log Out
Log Out
End Execution
End

Fig. 2. Flowchart of the System

VII. RESULTS

A. User Login

Users can register here and create an account, after which they can log in and upload files. This is shown in Fig.3

B. Admin Login

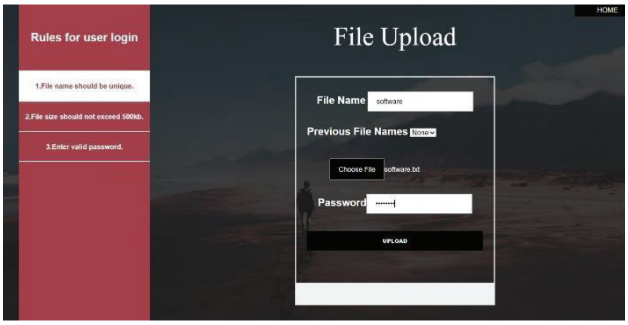
Admin doesn't need to establish an account because their credentials are always the same. Admin credentials can be used to log in.

C. Invalid login

We've included login validation so that, upon entering the ID and password, it will check against the credentials stored in the database; if the credentials are incorrect, it will indicate that the login attempt was unsuccessful.

D. Uploading Document

The user will be redirected to this page after successfully logging in, where they can upload a document for comparison. The document will be displayed alongside the previously available list of files, preventing file name conflicts. Once the password has been entered correctly, the file will be uploaded, and the similarity score will be monitored until the deadline has passed. Shown in fig.3



File Upload

Rules for user login

1.File name should be unique.

2.File size should not exceed 500kb.

3.Enter valid password.

File Name: software

Previous File Names: None

software.txt

Password: ••••••••

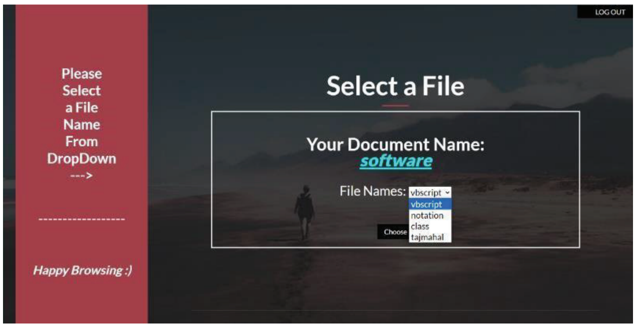
Fig. 3. Document Uploading

E. Admin Login

The administrator can view all available data, including a list of users and the data they have uploaded, a list of files uploaded by users and the content of those files both before and after preprocessing, when they check in before the deadline ends.

F. User login (After Deadline)

When the deadline expires, cosine similarity and latent semantic analysis will be computed. The user will then be redirected to the file selection page where they can select the file they wish to compare, and upon logging in, they will be able to view their score in a graphical format based on words and meaning. Steps shown in fig.4,5,6&7.



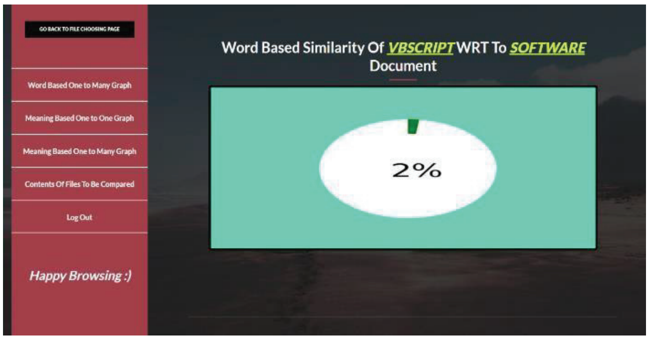
Select a File

Please Select a File Name From DropDown --->

Your Document Name: software

File Names: vbscript

Fig. 4. File Selection

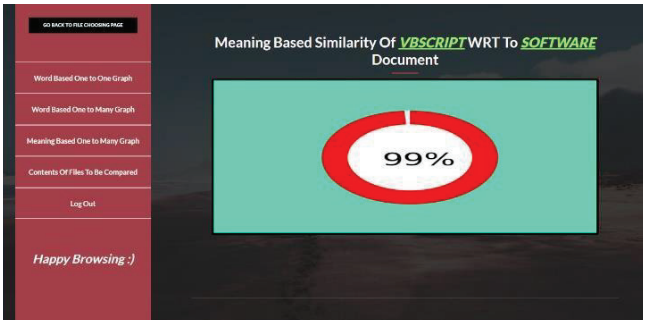


Word Based Similarity Of VBSCRIPT WRT To SOFTWARE Document

2%

GO BACK TO FILE CHOOSING PAGE
Word Based One to Many Graph
Meaning Based One to One Graph
Meaning Based One to Many Graph
Contents Of Files To Be Compared
Log Out
Happy Browsing :)

Fig. 5. (CS) Similarity Score

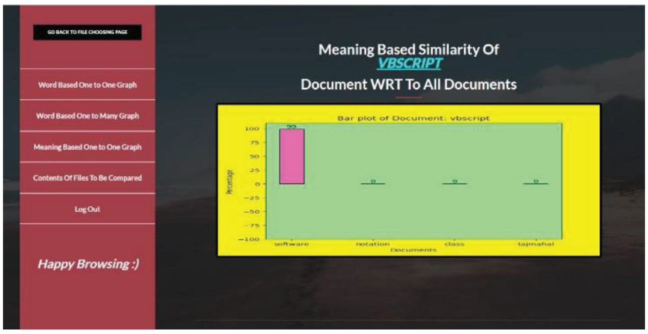


Meaning Based Similarity Of VBSCRIPT WRT To SOFTWARE Document

99%

GO BACK TO FILE CHOOSING PAGE
Word Based One to One Graph
Word Based One to Many Graph
Meaning Based One to Many Graph
Contents Of Files To Be Compared
Log Out
Happy Browsing :)

Fig. 6. Similarity Score (LSA)



Meaning Based Similarity Of VBSCRIPT Document WRT To All Documents

GO BACK TO FILE CHOOSING PAGE
Word Based One to One Graph
Word Based One to Many Graph
Meaning Based One to One Graph
Contents Of Files To Be Compared
Log Out
Happy Browsing :)
Bar plot of Document: vbscript
Percentage
100
75
50
25
0
−25
−50
−75
−100
99
0
0
0
software
notation
class
tajmahal
Documents

Fig. 7. Similarity Score with All Documents

G. Admin Login (After Deadline)

The admin will log in and view all the data, including the uploaded document, the list of registered users, the cosine similarity and latent semantic analysis of all the files in the dataset, once the deadline has passed and the similarity calculation has been completed.

VIII. CONCLUSION AND FUTURE WORK

This project's primary goal is to give the user an accurate similarity score based on both word and meaning similarity. In order to make the results easier to read and comprehend, it also emphasizes presenting the data in graphical form. To accurately complete the task, this system makes use of techniques like Latent Semantic Analysis and algorithms like Cosine Similarity. The systems need a document, which is processed further to get the desired outputs. By comparing documents related to their area, the system can be highly helpful to the business in reducing internal redundancy.

An offline tool such as ours allows documents to be compared only within the document's range. Since this tool doesn't require an internet connection, data privacy is guaranteed. The document has benefits. which document they wish to compare and enhance their document's thesis. For internal use, it would be preferable to check similarity only within the confines of the document. The figure 8 illustrates, the computation time increases as the number of documents increases. We can alter this situation by enhancing system performance so that computation times will be shortened, as this offline tool is entirely dependent on system configuration.

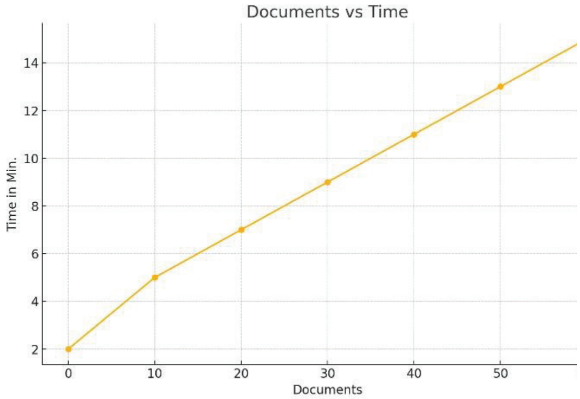


Fig. 8. Documents by Time

Even though there are other tools of this kind, the company can utilize this one internally as it allows them to compare or use documents related to their domain, which produces relevant results.

REFERENCES

[1] Shaikh and A. Kumar, "A Comparison between Syllabus of AICTE and various Universities - An NLP Based Approach," 2021 2nd International Conference for Emerging Technology (INCET), Belagavi, India

[2] S. P. and A. P. Shaji, "A Survey on Semantic Similarity," 2019 International Conference on Advances in Computing, Communication and Control (ICAC3), Mumbai, India

[3] T. Islam, M. Hossain and M. F. Arefin, "Comparative Analysis of Different Text Summarization Techniques Using Enhanced Tokenization," 2021 3rd International Conference on Sustainable Technologies for Industry 4.0 (STI), Dhaka, Bangladesh

[4] M. Saeed and A. Y. Taqa, "An Intelligent Approach for Semantic Plagiarism Detection in Scientific Papers," 2022 8th International Conference on Contemporary Information Technology and Mathematics (ICCITM), Mosul, Iraq

[5] F. Ahmad and M. Faisal, "Comparative Study of Techniques used for Word and Sentence Similarity," 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India

[6] Viji and S. Revathy, "Semantic Similarity Detection from text document using XLNet with a DKM- Clustered Bi-LSTM Model," 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India

[7] Srivamsi, O. M. Deepak, M. D. A. Praveena and A. Christy, "Cosine Similarity Based Word2Vec Model for Biomedical Data Analysis," 2023 7th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India

[8] P. Kherwa and P. Bansal, "Latent Semantic Analysis: An Approach to Understand Semantic of Text," 2017

[9] International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), Mysore, India,

[10] L. Shkurti, J. Ajdari, F. Kabashi and V. Fusa, "PlagAL: Plagiarism detection system for Albanian texts," 2021 10th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro