ISSN NO: 2770-0003
Date of Publication: 26-04-2023

Check for Plagiarism Using Text Mining

Umidakhon Zukharova
Bachelor of Technology in Computer Science Engineering
Abubakir Akbarov
Bachelor of Technology in Computer Science Engineering

Abstract: The internet provides answers to all queries in the modern world. Therefore, anybody may quickly copy and utilize material from the web at any time. Plagiarism is what is meant by this. It is currently expanding. Typically, when someone plagiarizes a piece of writing, they rephrase, duplicate, and omit sources. Plagiarism is difficult to identify since individuals often paraphrase texts rather than copying them verbatim. The Apache Lucene plagiarism detector has been utilized. The original document is first indexed, and the copied document is then compared to a collection of previously stored documents using cosine similarity.

Keywords: Indexing, Apache Lucene, Cosine similarity, and Plagiarism

1. Introduction

Any document is really just a collection of words, key phrases, and other terminology. Any of the aforementioned bases may be used to analyze a piece of writing. Nowadays, it is quite simple to steal someone else's ideas and work or claim credit for it. This behavior is known as plagiarism. Copying from websites or other people's assignments is a relatively widespread practice in academia. Therefore, it's important to detect or prevent plagiarism among pupils. One needs a plagiarism detection program or checker for this. However, this does not imply that students shouldn't use other people's works as references despite their fear of being accused of plagiarism. In actuality, it is an excellent technique to expand your knowledge or do research in your specific field, but the student must provide due acknowledgment to the sources' original authors and the pertinent mentioned references. Text mining essentially involves arranging the provided text before looking for or evaluating any patterns. The primary goal is to identify differences in the other person's document's writing style. Text mining is used to assess how similar the text content is, which will eventually result in plagiarism detection. In order to integrate unstructured data with structured data and compare the results, it essentially turns the words and phrases that make up that data into numerical values. Additionally, it does not require installation and automatically updates the user. The interface is inferior, and the character limit is 1000 per line. There are also paid versions available.[3] Additionally, it does not require installation and automatically updates the user. The interface is inferior, and the character limit is 1000 per line. There are also paid versions available.[3] Additionally, it does not require installation and automatically updates the user.

The interface is inferior, and the character limit is 1000 per line. There are also paid versions available.[3]

The detection of plagiarism is possible both manually and with the use of software. Manual detection requires a great deal of comparison and intelligence since it involves comparing several papers to the original. Tools make it simpler to determine if the material has been unlawfully copied, allowing for a quick comparison of the document with a large number of research papers and other online data. Several tools are already available for this reason; the article currently uses a text plagiarism checker.

2. Related Work

There has been researching done on many methods and techniques used to identify plagiarism.

2.1 Types

There are three primary categories of software-based plagiarism detection tools:

1. Textually Based
2. Based on attribute-oriented code
3. Code-based structure-oriented system

ISSN NO: 2770-0003
Date of Publication: 26-04-2023

Textually Based

One may measure word and phrase frequency counts to determine whether or not a certain text contains any instances of plagiarism. A greater rate of similarity is indicated by a higher word count frequency. Only when the text is completely copied and unaltered is this kind helpful. The four main processes are collection, analysis, conformation, and inquiry. [1,2]

Based on attribute-oriented code

This just evaluates the fundamental characteristics of the code. By comparing these traits' differences, one may determine how similar two things are. Despite not checking the other variables, this type is not particularly helpful since anybody may easily replicate a piece of code by just changing the variable name. Additionally, it is quite challenging to check for plagiarism in huge code since it will take a long time to do line-by-line analysis.[2]

Structure-oriented code-based

It essentially combines the first two methods described. It takes care of both textual and structural issues. Minor adjustments are also made to variables, comments, and the whole structure. Therefore, it is better than the other two specified methods tools.[1,2].

2.2 Currently available tools

Plagiarism Checker By Smallseotools

This is an instrument made available online by smallseotools.com. You need only copy and paste the text into the provided area. Click the icon to verify for duplicate content if the copied text turns scarlet, it is considered to be plagiarized.

PlagScan

Additionally, it does not require installation and automatically updates the user. The interface is inferior, and the character limit is 1000 per line. There are also paid versions available.[3].

PlagTracker

It is utilized to detect academic plagiarism and is significantly quicker. Although it is not completely correct, it provides information on the sources of the material that was copied. Grammar checking is another function available in the premium edition.[6]

Viper

It scans the copied paper through more than 10 billion academic and other web sources and is completely free. Although it is completely free, downloading the program is necessary. Only users of Microsoft Windows are permitted to utilize it.[6]

QueText

It has a user-friendly UI and is completely free. There is no need to register or download any software. To verify the whole document at once, however, you must copy and paste the text since you are unable to upload the file here.[6] **Plagium**

It will offer you a URL and can only examine up to 250 characters at once, but even though it's free, you must first join up to use the tool.[3]

Turnitin

It's a useful text-based resource for classroom instructors and students alike. It may function within the body as well as outside. According to statistical evidence, it is the best detecting tool available online.[7] There are many more tools which are available for the same.

2.3 Existing Approaches

Existing methods can be categorized as either extrinsic or intrinsic. Extrinsic methods consist primarily of comparing dubious documents to authentic ones. Several comparison approaches have thus been proposed. In contrast to extrinsic methods, intrinsic ones investigate characteristics such as linguistic ones without comparing them to external documents. It focuses primarily on lexical characteristics, syntactic characteristics such as word frequency, and structural characteristics such as average paragraph length.[5]

The similarity score is primarily determined by various metrics. On the basis of the number of documents involved, metrics can be broadly divided into two categories: singular and multidimensional.[7]

Different comparison strategies are:-

ISSN NO: 2770-0003
Date of Publication: 26-04-2023

The main goal of string matching processes is to find the longest pair of text strings that are identical. The data is then determined to be plagiarized if a certain threshold is surpassed. Although there are other techniques, including suffix trees and suffix arrays, it is challenging to identify masked plagiarism.

Vector space based, it essentially takes into account a collection of phrases that have been taken from the whole manuscript for similarity purposes. For the similarity, the well-known cosine measurements are used. To look for word synonyms, semantic relationships, and other words that have changed, one might utilize more advanced functions.

Fingerprinting is used to carry out local comparisons. Their goal is to pick out a number of different substrings from the text. Fingerprints are the collection of substrings, while the individual substrings are known as minutiae. To facilitate comparison, hash functions are utilized to transform the data into a string format. Each specific detail requires its own query to check against the index.**TF-IDF and LSI**

Longest common subsequence finding the longest common subsequence between two subsequences. It differs from typical substrings in that the subsequence need not occupy consecutive positions with the original document. Using prefixes, the document is subdivided into smaller subproblems that make its comprehension easier. All prefixes are stored in a table, and by comparing the scores, the longest common subsequence is determine

3. Proposed Work

This approach tries to make a plagiarism checker by
combining Apache Lucene and cosine similarity to analyze
whether the text has been copied.

3.1 Apache Lucene

It is Java-based open-source software created by Doug Cutting. It is appropriate for any program that has to be able to search and do indexing. It may be used for both single-site and local searches.It has been integrated into our software using the Lucene-core-3.6.0 jar file.

The Apache Software Foundation supports it. Both open-source and for-profit software may utilize it.

Lucene Indexing

Indexing is used to boost performance and enhance speed. The data is then searched using the provided query. If indexing is not done, the document will be scanned in bulk, which will use a lot of computational resources and take hours.[4]

Generally speaking, Lucene utilizes two steps:

1. The page is indexed in Lucene.
2. Parse the question and use the index created before to get the appropriate response.

One may build an object of the type IndexWriter to do indexing. Additionally, it is used to supplement the current index with fresh index entries, or the entries of new documents.

There are several sorts of analyzers available for it to parse the input into indexed tokens or keywords. The implementation goal has been served by the usage of a standard analyzer. Plain English wording should be used for the data.

Basically, there are four categories of analyzers:

1. StandardAnalyzer: It is a general-purpose analyzer.
2. WhitespaceAnalyzer: Only whitespace is used to separate tokens.
3. StopAnalyzer: This feature eliminates words that aren't utilized for indexing.
4. The Snowball Analyzer analyzes all word roots.
5. Relating to concealing is like hiding

Although StandardAnalyzer performs a good job, anybody can utilize an analyzer.

The document may then be added using the Document class to the newly constructed index. Three factors are needed: storage flag, field value, and field name. The third option indicates whether the index value should be preserved or discarded. Afterwards, query indexing may be used for text search.

3.2 Cosine similarity

There are several techniques for detecting plagiarism, each of which uses a different algorithm. Tools for detecting plagiarism may be created using a variety of algorithms. One such method is cosine similarity. In its

ISSN NO: 2770-0003
Date of Publication: 26-04-2023

most basic form, cosine similarity is only a mathematical idea, yet it has several applications in Relevance Ranking, Text Mining, and Information Retrieval.

Cosine similarity calculates the degree of similarity between two papers. In essence, it calculates the sine of the angle between the supplied documents. For example, two papers with the same orientation will have a cosine similarity of 1, which indicates that one of those documents is entirely copied from the other. In this case, the cosine angle of provided documents represents a judgment of orientation, not magnitude.

At 90 degrees, two papers will have a similarity of 0, which indicates that they are both wholly distinct. As a result, cosine similarity provides an effective way to compare two papers' similarities:

Now it is possible to ascertain how the cosine similarity algorithm operates by considering documents as vectors.

The Euclidean dot product formula is utilized to determine the cosine angle.

$$a.\ b = a\ |b|cos\theta$$

Given two vector attributes A and B, cosine similarity, cos(θ), can be represented using the dot product, and its magnitude is given as

$$\text{Similarity} = \cos.\ B/|\ A\ |B|| = ni{=}1Ai \times Bi\ /\ ni\ Bi$$

The resulting similarity will range from -1, which represents the exact opposite, to 1, which represents the exact same, with 0 representing orthogonality and values in between representing the intermediate similarity of the text. [4]

Here are two brief texts for comparison:

1. Julien favors me more than Linda does.
2. Jany adores me more than does Julien.

To determine the degree of similarity between these two texts, one can create an inventory of the characters that appear in both:

Me, Julien, loves, Linda, than, more, likes, Jany. Therefore, it is possible to ascertain the frequency of text appearances: -

me 2 2
Jany 0 1
Julien 1 1
Linda 1 0
loves 0 1
likes 2 1
more 1 1
than 1 1

Essentially, one only needs to be aware of the frequency counts of the word and the vectors that are formed.

The two vectors are:-
a: [2, 1, 0, 2, 0, 1, 1, 1]
b: [2, 1, 0, 2, 0, 1, 1, 1]

The cosine of their angle is approximately 0.822. In this instance, the angle is 35 degrees.

4. Implementation

The following are the screenshots of the software made:-

Fig. 1 depicts the first page of the plagiarism checker's user interface, which provides an overview of the tool and its graphical user interface. The first tab is the about tab, which provides information about the initiative.

ISSN NO: 2770-0003
Date of Publication: 26-04-2023

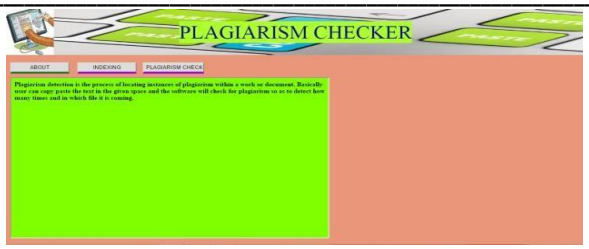

Fig 1: The first page , the user interface

Fig. 2 depicts the indexing of the plagiarism detector, which indexes the documents to which the plagiarized work will be compared. It stores the index files in a folder that will be specified in the output.

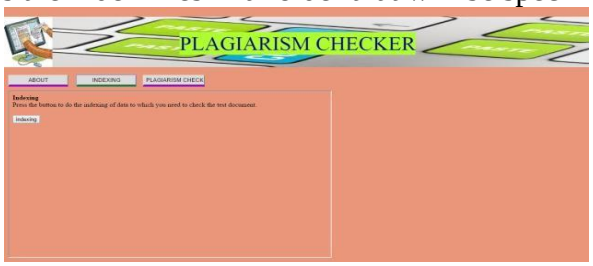

Fig 2 :The indexing page to perform indexing

Fig. 3 depicts the result of indexing, which is a successful display if the index files are saved in a folder.

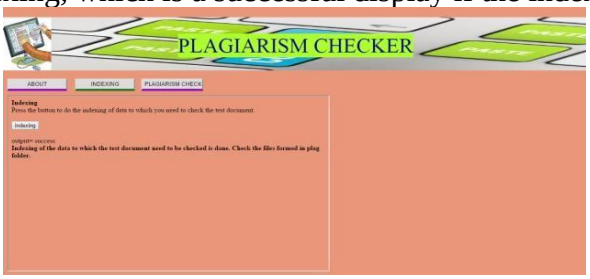

Fig 3: After indexing is done

The files that are kept in the folders after indexing are shown in Fig. 4.

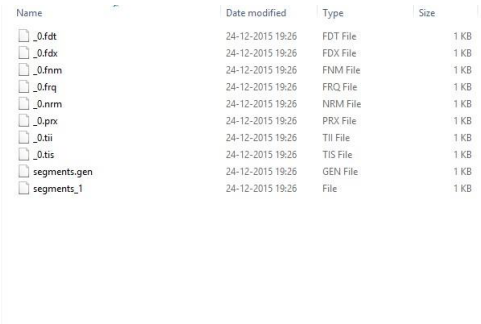

Fig 4: Files created After Indexing

Fig 5 shows the plagiarism check button that is the text area and the submit button the text which one need to copy and paste.

ISSN NO: 2770-0003
Date of Publication: 26-04-2023

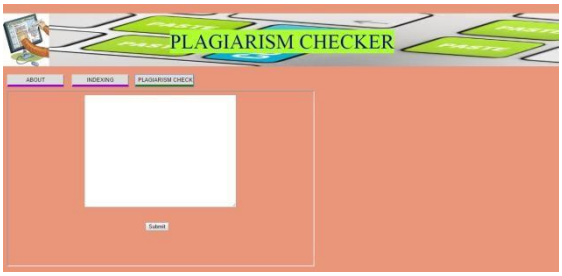

Fig 5: Plagiarism check button

Fig 6 shows the output of the plagiarism check that is tells the number of documents and the match score of the word checked.

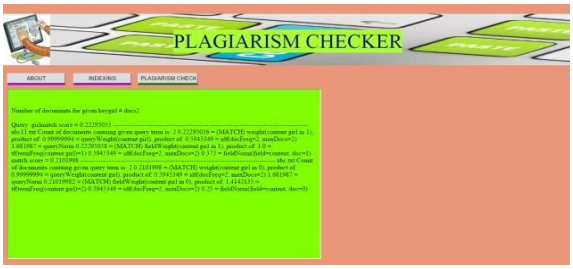

Fig 6: Output of plagiarism check

5. Conclusion The two basic things used are:-

• Indexing is done on the documents using Apache Lucene to make searching easier.
• Searching based on the search term used to look for text that has been put into the text box.

The program that has been developed can look for terms that have been copied and provide a match score based on cosine similarity. The material that has to be examined may be readily copied and pasted in the designated space

The future work can be:

□□Checking for structure-oriented code is another important step.
□□By using Wordnet, this will also find synonyms.
□□Using an effective string matching method can further cut down on time and improve efficiency.

References

1. International Journal of Innovative Research in Advanced
2. Fig 5: Plagiarism check button Engineering (IJIRAE) ISSN: 2349-2163 Volume 1 Issue 7, August 2014
3. Ahmad Gull Liaqat & Aijaz Ahmad ,"Plagiarism Detection in Java Code ",Linnaeus University, June 2011
4. Asim M. El Tahir Ali, Hussam M. Dahwa Abdulla, and V´aclav Sn´aˇsel ,"Overview and Comparison of
5. Plagiarism Detection Tools" 161{172, ISBN 978-80-
6. 248-2391-1., 2011
7. Daniele Anzelmi, Domenico Carlone, Fabio Rizzello,
8. Robert Thomsen, D. M. Akbar Hussain,"Plagiarism Detection Based on SCAM Algorithm", Proceedings of the International MultiConference of Engineers and Computer Scientists, March 2011
9. Romans Lukashenko, Vita Graudina, Janis
10. Grundspenkis, "Computer-Based Plagiarism Detection Methods and Tools: An Overview", International Conference on Computer Systems and Technologies - CompSysTech'07, 2007
11. Reena Kharat, Preeti M. Chavan, Vaibhav Jadhav, Kuldeep Rakibe,"Semantically Detecting Plagiarism for Research Papers", International Journal of
12. Engineering Research and Applications (IJERA), May-Jun 2013