Plagiarism Checker: Text Mining

Anu Saini
Assistant Professor
Maharaja Surajmal Institute of Technology

Ankita Bahl
Undergraduate Student
Maharaja Surajmal Institute of Technology

Supriya Kumari
Undergraduate Student
Maharaja Surajmal Institute of Technology

Mitali Singh
Undergraduate Student
Maharaja Surajmal Institute of Technology

ABSTRACT
In today's world internet is the answer to every question. So at any time one can easily copy the content from web and use it. This is known as plagiarism .It is growing now days. Usually in plagiarism people reword the documents, copy them, do not give references. It is difficult to detect plagiarism as people rephrase the text do not copy it directly. To detect plagiarism apache lucene have been used. Firstly indexing of the original document is done and then used cosine similarity to compare the plagiarised document with set of documents which are there saved previously.

Keywords
Apache Lucene, Indexing, Cosine similarity, Plagiarism

1. INTRODUCTION
Any document is basically a set of words, keywords, and some terms. One can analyse a document on any of the above mentioned basis. Now it is so easy to owe someone's ideas and work or take the credit of other's work. The term used for this act is Plagiarism. In academics it is very common to copy others assignment or from any site. So there is a need to identify or stop students from plagiarism. For this one need the plagiarism tool or plagiarism checker. However this does not mean that from the fear of plagiarism the student should not take view in others work for the references. In fact it is a good way to improve your knowledge or does research in your particular domain, but the student must credit the origin of sources and the relevant cited reference.

Text Mining is actually structuring the given text and then finding or analyzing the patterns in it. The main aim is to find deviation in the style of writing of the document written by the other person. The similarity in the text document is measured using text mining which will ultimately lead to plagiarism detection. It basically converts words and phrases which are the part of unstructured data into numerical value so as to link it with structure data and then compare it for further action.

Plagiarism can be detected manually or using software assisted tools. Manual detection requires a lot of comparison and good intelligence as one need to compare many documents with the original one. Tools make it easier to check whether the data has been copied illegally so as in seconds one can easily compare the document with plenty of research papers and other data from internet. Several tools are already available for this purpose, the paper implements text plagiarism checker for now.

2. RELATED WORK
Different types of tools and currently existing approaches used for plagiarism detection have been studied.

2.1 Types
There are broadly three types of software based plagiarism detection tools

1. Text Based
2. Attribute-oriented code-based
3. Structure-oriented code-based system

Text Based
To check whether a particular document is plagiarized or not one can measure frequency counts on words and sentences. The higher frequency of word count indicate higher rate of similarity. This type is useful only when the text is fully copied without minor alterations. It basically consist of four stages collection, analysis, conformation and investigation. [1,2]

Attribute-oriented code-based

In this the main key properties of the code and appraises them only. The similarity is judged by measuring the difference between these attributes. Although this type is not very useful as it does not check the other variables as someone can easily copy a code by just changing the variable name. Also it is very difficult to check plagiarism for large code as it will take long time as line by line checking is done.[2]

Structure-oriented code-based

It is basically the combination of above two mentioned techniques. Both textual and structural things are taken care of. The minor changes are also taken care of like variable changes, comments, and whole structure. So it is more beneficial than the other two mentioned techniques tools. [1,2].

2.2 Currently available tools
PLAGIARISM CHECKER by SmallSEOTools
This is an online tool provided by smallseotools.com. You just simply need to copy and paste the text in the box given. Click on the button check for plagiarism if the text which you have pasted becomes red then it is plagiarized.

PlagScan
It also does not require any installation and updates the user continuously. The interface is not as good and limit is 1000 character at a time. Paid versions are also available.[3]

PlagTracker
It is much faster and is used for academic plagiarism detection. It offers details about from where the text has

been copied, although it is not 100% accurate. The paid version also has the feature of grammar check.[6]

Viper
It is purely free and scans the copied document through more than 10 billion academic and other online sources. It requires the software to be downloaded although it is 100% free. It can be used only by the Microsoft windows users.[6]

QueText
It's 100% free and have a user friendly interface. No need to download the software or create an account. But here you cannot upload the file over here to check whole document at one go you need to copy paste the text.[6]

Plagium
It can check only upto 250 characters only at a time, it will give url also but for using this tool you need to signup first although its free of cost.[3]

Turnitin
It is text based tool which is designed for both teachers and students. It operates both intra-corpally and extra-corpally. It is the web leader in detection tool according to analytical data.[7]

There are many more tools which are available for the same.

2.3 Existing Approaches

Existing methods can be classified in to two categories extrinsic and intrinsic. **Extrinsic methods** basically include comparison of suspicious document with the genuine works. For that several comparison methods have been suggested. **Intrinsic methods** they are opposed to extrinsic one, they examine features like linguistic one without doing comparison with the external documents. It basically aims at lexical features, syntactic features-word frequencies, structural features like average length of the paragraph.[5] Similarity score is basically based on different metrics. Broadly the metrics, on the basis of number of documents involved can be classified in to two categories that is singular or multi-dimensional metrics.[7]

Different comparison strategies are:-
String matching procedures, they basically aims at identifying the longest pair of identical text string. Then a threshold is decided, if that threshold is exceeded then the data is plagiarized. Several methods like suffix tree, suffix arrays are used, although it is difficult to detect disguised plagiarism

Vector space based, it basically consider a set of terms which has been extracted from whole document for the similarity purpose. The well-known cosine measures are used for the similarity. One can use more complex functions to check for word synonyms, semantic relations and other terms also which are changed.

Fingerprinting is used to perform local similarity. They aim to select multiple substrings from the text. The set of the substrings is called fingerprints and its elements are called minutiae. Hash functions are used to convert the minutiae into string type which can be easily compared. A query is used for each minutiae to compare it with the indexed document.
TF-IDF and LSI is basically an encoding technique based on inverse document frequency, a weighted matrix is created, Basically TF that is term of frequency is the weight which is assigned to each token it is local one and IDF is global one which calculates inverse frequency of tokens from research papers in database. Then LSI is used which use single value decomposition to replace short vectors with original data vectors and then display result in descending order.[8]

Longest common subsequence hereby we find the longest subsequence common to the two subsequences. It differs from common substrings, here it is not necessary that subsequence occupy consecutive positions with that of original document. The document is divided into smaller sub problems which make it easy by using prefixes. All the prefixes are stored in a table and using this longest common subsequence is found by comparing the scores.

3. PROPOSED WORK

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

3.1 Apache Lucene

It is open source software written in java by Doug cutting. It is suitable for any software which requires indexing to be done and needs searching capability. It can be used for local as well as single site searching.Lucene-core-3.6.0 jar file has been used for implementing it in our program.

It is supported by apache software foundation. It can be used in both open source and commercial programs.

Lucene Indexing
To optimize the speed, to increase the performance indexing is used. Afterwards data is searched according to the given query. If one don't do indexing it will scan the document in bulk which will require a lot of computing power, which would take hours.[4]

Broadly lucene uses two steps:-
1. Lucene indexing is done of the document
2. Parse the query and look up the index formed earlier to answer the query accordingly.

To perform indexing one can create an object of IndexWriter class. It is also use to add new index entries that is the entries of the new documents to the existing index.

Analyzers are used to parse the data into indexed tokens or keywords there are different types of analyser available for it. Standard analyser has been used for the implementation purpose. The data should be plain English text.

There are four types of analyzers basically:-
1. StandardAnalyzer:- It is a analyser used for general purpose
2. WhitespaceAnalyzer:- It separates tokens using whitespace only
3. StopAnalyzer:- It removes the words which are not used for indexing
4. SnowballAnalyzer:-it works on all word roots
5. like hide relates to hiding

One can use any analyzer although StandardAnalyzer do well job.

Then one can add the document to the created index using Document class. It requires three parameters field name,

field value, storage flag. The third parameter shows whether the value indexed need to be saved or it can be discarded. Afterwards one can perform text search using query indexing.

3.2 Cosine similarity

There are different plagiarism detection tools available which work on different algorithm. There are several algorithm to create plagiarism detecting tools. Cosine similarity is one of such algorithms. Cosine similarity in a narrow sense is just a mathematical concept, but this concept has various applications in Information Retrieval, Text Mining and Relevance Ranking.

Cosine similarity measures a similarity factor between two documents. It basically determines the cosine of the angle between the given documents. Here, the cosine angle of given documents means a judgement of orientation, not magnitude, for example two documents with same orientation will have a cosine similarity of 1,which shows that one of those documents is completely plagiarized to the other.

Similarly, two documents at 90^o will have a similarity of 0 , that means both documents are purely unique. Hence, cosine similarity gives a useful measure of how similar two documents are.

Now how cosine similarity algorithm works can be determines by considering documents as vectors.

Euclidean dot product formula is used to find cosine angle

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

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

Similarity=$cos\theta = A.B/||A||B||$

$=\sum_{i=1}^{n} Ai \times Bi \,/\sqrt{\sum_{i=1}^{n}(Ai) * (Ai)} \times \sqrt{\sum_{i=1}^{n}(Bi) * (Bi)}$

The resulting similarity will range from −1 which means exact opposite, to 1 which means exactly the same, and when 0 it indicates orthogonality, and in-between values indicates the intermediate similarity of the text. [4]

Here are two very short texts to compare:

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

To know how similar these two texts are one can make the list of the characters which are there in two lines:-

Me, Julien, loves, Linda, than, more, likes, Jany. So afterwards one can count the number of time text comes:-

me 2 2

Jany 0 1

Julien 1 1

Linda 1 0

loves 0 1

likes 2 1

more 1 1

than 1 1

Basically one just needs to know about the counts of how much time the word occurs, one should just know the vectors formed.

The two vectors are:-

a: [2, 1, 0, 2, 0, 1, 1, 1]

b: [2, 1, 1, 1, 1, 0, 1, 1]

The cosine of the angle between them is about 0.822. In this case the angle comes out to be 35 degree.

4. IMPLEMENTATION

The following are the screenshots of the software made:-

Fig 1 shows the first page of the plagiarism checker which tells us about it, gives the first look of graphical user interface. The first tab is about tab which tells about the project.

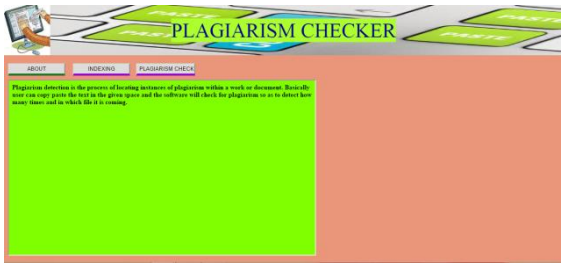

Fig 1: The first page , the user interface

Fig 2 shows the indexing of the plagiarism checker which does the indexing of the documents to which one will compare the plagiarised work. It store the index files in a particular folder which will be mentioned in the output.

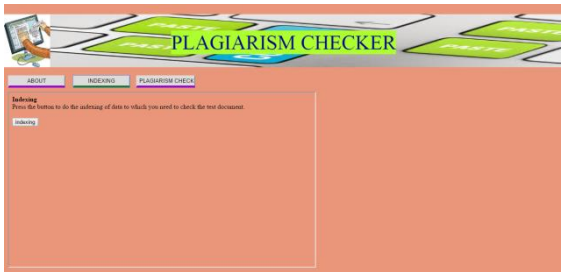

Fig 2 :The indexing page to perform indexing

Fig 3 shows what happens when indexing is done that is it gives success display if the index files are stored in a folder.

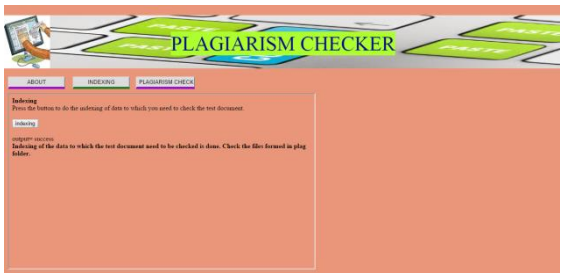

Fig 3: After indexing is done

Fig 4 shows the files which are stored in the folders after indexing.

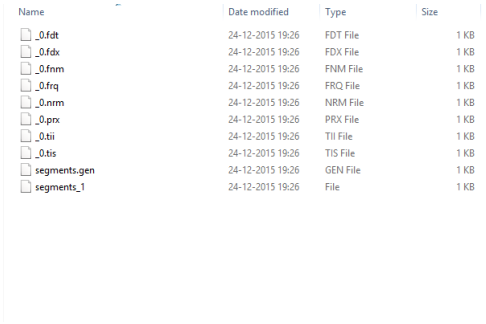

Fig 4: Files created After Indexing

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

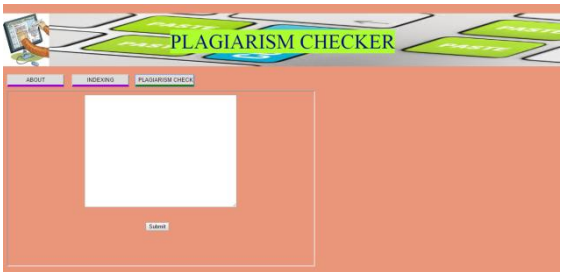

Fig 5: Plagiarism check button

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

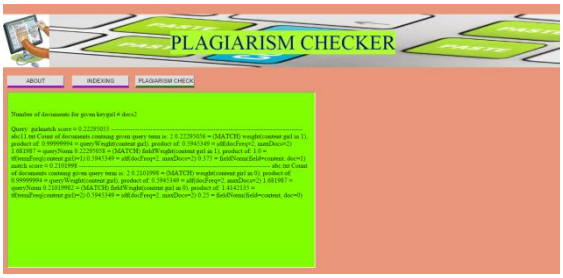

Fig 6: Output of plagiarism check

5. CONCLUSION

The two basic things used are:-

- Apache lucene is used to perform indexing on the documents so as to make search easy.
- Searching according to the query that is to search for the text which has been pasted in the text area.

The software prepared is able to search words that are plagiarised and give the match score according to cosine similarity. One can easily copy the content which is to be checked and paste it in the provided area.

The future work can be:

- Checking for code also that is structure oriented detection.
- Implementing Wordnet also this will detect synonyms as well.
- Applying efficient string matching algorithm which will further reduce the time and increase the efficiency.

6. REFERENCES

[1] S.A.Hiremath and M.S.Otari ,"Plagiarism Detection-Different Methods and Their Analysis: Review", International Journal of Innovative Research in Advanced Engineering (IJIRAE) ISSN: 2349-2163 Volume 1 Issue 7, August 2014

[2] Ahmad Gull Liaqat & Aijaz Ahmad ,"Plagiarism Detection in Java Code ",Linnaeus University, June 2011

[3] Asim M. El Tahir Ali, Hussam M. Dahwa Abdulla, and V´aclav Sn´aˇsel ,"Overview and Comparison of Plagiarism Detection Tools" 161{172, ISBN 978-80-248-2391-1., 2011

[4] Daniele Anzelmi, Domenico Carlone, Fabio Rizzello, Robert Thomsen, D. M. Akbar Hussain,"Plagiarism Detection Based on SCAM Algorithm", Proceedings of the International MultiConference of Engineers and Computer Scientists, March 2011

[5] Bela Gipp Norman Meuschke ,"Citation Pattern Matching Algorithms for Citation-based Plagiarism Detection: Greedy Citation Tiling, Citation Chunking and Longest Common Citation Sequence", Mountain View, CA, USA, September 2011

[6] http://elearningindustry.com/top-10-free-plagiarism-detection-tools-for-teachers

[7] Romans Lukashenko, Vita Graudina, Janis Grundspenkis, "Computer-Based Plagiarism Detection Methods and Tools: An Overview", International Conference on Computer Systems and Technologies - CompSysTech'07, 2007

[8] Reena Kharat, Preeti M. Chavan, Vaibhav Jadhav, Kuldeep Rakibe,"Semantically Detecting Plagiarism for Research Papers", International Journal of Engineering Research and Applications (IJERA), May-Jun 2013