Plagiarism Detection And Visual Inspection Of Data

Pranoti Nage[1], Priyanka Sharma[2],Vaishali Salvi[3], Reena Somani[4] , Sejal D'mello[5]

[1]*(Information Technology, Mumbai University, India)*
[2]*(Information Technology, Mumbai University, India)*
[3]*(Information Technology, Mumbai University, India)*
[4]*(Information Technology, Mumbai University, India)*
[5]*(Information Technology, Mumbai University, India)*

Abstract: *Digital documents are vulnerable to being copied. Most existing copy detection prototypes employ an exhaustive sentence-based comparison method in comparing a potential plagiarized document against a repository of legal or original documents to identify plagiarism activities. This approach is not scalable due to the potentially large number of original documents and the large number of sentences in each document. Furthermore, the security level of existing mechanisms is quite weak; a plagiarized document could simply by-pass the detection mechanisms by performing a minor modification on each sentence. In this paper, we propose a copy detection mechanism that will eliminate unnecessary comparisons. The system will parse the data to compare against the data obtained from web to detect plagiarism using Hadoop in the backend to increase the efficiency of the system..*

Keywords: *Plagiarism,Hadoop, High charts,Tokens,Comparator*

I. Introduction

Now-a-days, internet is one of the most important factors in life. Plagiarism has become a worldwide problem and is increasing day by day. This problem is getting worse mainly because of the increase in the volume of on-line publications. Relying only on exact word or phrase matching for plagiarism detection is not sufficient now. People have started paraphrasing or rearranging words to give a new look to their sentences and thus declare themselves as authors of the material. Using Plagiarism Detection Techniques we can compare a given material with any target material which is either a particular document or in a repository. Different techniques used in the Plagiarism Detection algorithms are discussed in detail here. Here I have given more emphasis on source code related plagiarism. A few case studies show that detection can be done within a large repository. The efficiency and time of the output depends on the algorithms used.

Definition of Plagiarism:

Plagiarize according to the Merriam-Webster Online Dictionary is:

- To steal and pass off (the ideas or words of another) as one's own
- To use (another's production) without crediting the source
- To commit literary theft
- To present as new and original an idea or product derived from an existing source

The expression of original ideas is considered intellectual property and is protected by copyright laws, just like original inventions. Almost all forms of expression fall under copyright protection as long as they are recorded in some way (such as a book or a computer file). In other words, plagiarism is an act of fraud. It involves both stealing someone else's work and lying about it afterward [1].

The following are considered as plagiarism:

- Turning in someone else's work as your own
- Copying words or ideas from someone else without giving credit
- Failing to put a quotation in quotation marks
- Giving incorrect information about the source of a quotation
- Changing words but copying the sentence structure of a source without giving credit
- Copying so many words or ideas from a source that it makes up the majority of your work, whether you give credit or not

Plagiarism can be deliberate or accidental. [2] Figure 1 shows the range between Deliberate and Accidental Plagiarism. Deliberate plagiarism is done when a person's self-esteem is very low. The person, therefore, actually steals the property of somebody else and claims it to be his own. He might also hire somebody to do his work. Accidental plagiarism is done when somebody unknowingly cites a phrase or copies words without acknowledging the author of the material.

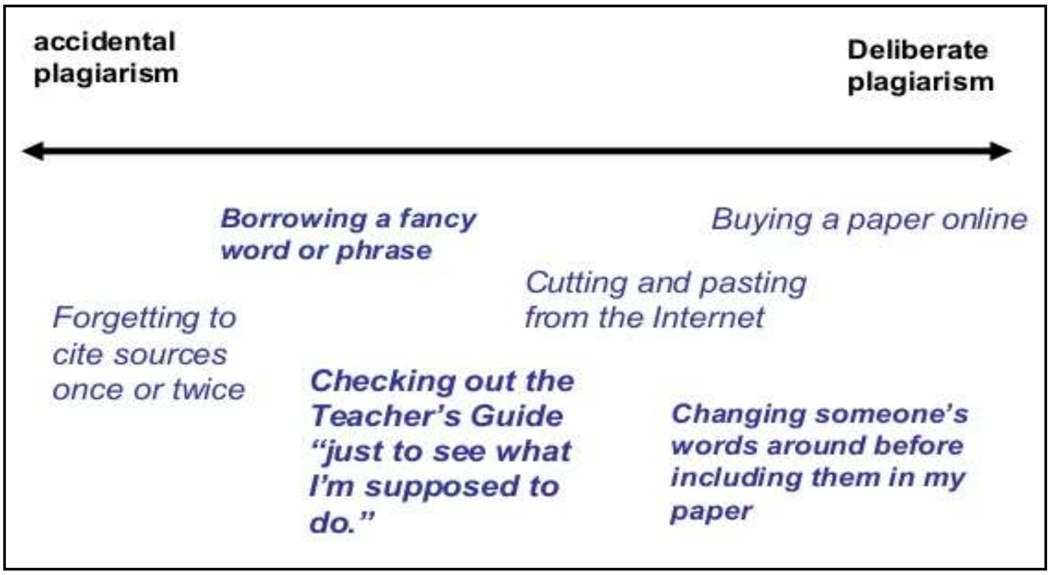
accidental plagiarism
Deliberate plagiarism
Borrowing a fancy word or phrase
Buying a paper online
Cutting and pasting from the Internet
Forgetting to cite sources once or twice
Checking out the Teacher's Guide "just to see what I'm supposed to do."
Changing someone's words around before including them in my paper

Figure 1: Difference between accidental and deliberate plagiarism

In the present age of computers and internet the attack of plagiarism has increased. Digitalized documents have only provided a means to increase plagiarism. This has led to an adverse effect on the learning criteria of youngsters today. They are depriving themselves of better learning opportunities and have become addicted to plagiarism in such a way that the true identity of the material is lost. This has also led to professional dishonesty in the job front. The rapid increase in the digital documents and the easy access to the material has led to increase in plagiarism. This has in turn decreased the chances of detecting plagiarism. Now it is more and more difficult to find plagiarized content. So currently we have started many methods and techniques to detect and to avoid plagiarism.

Avoiding Plagiarism

Plagiarism in all kinds of work has made people to sit and think regardingthe ways to avoid plagiarism. Mainly two methods exist to avoid plagiarism.

- Plagiarism prevention
- Plagiarism Detection

Plagiarism Prevention

- A collaborative effort should be made to recognize and to counter plagiarism at every level.
- We should educate students about the appropriate use and acknowledgement of all forms of intellectual material.
- Minimize the possibility of submission of plagiarized content while not reducing the quality and rigor of assessment.
- Installing highly visible procedures for monitoring and detecting cheating.

Plagiarism prevention is difficult to achieve and takes a long time to inculcate but the effects are long term.

Plagiarism Detection

Plagiarism can be detected manually or with the help of software manual detection takes more effort. Now the detection Techniques is software programming methods which are easier, simpler and faster to detect plagiarism.

II. Related Work

Most of the data, source codes are available on internet and easy to access which leads to copy them as it is, called plagiarism. [2]The earliest technique used to detect plagiarism isa four stage process defined byCulwin and Lancaster shown in figure 2.

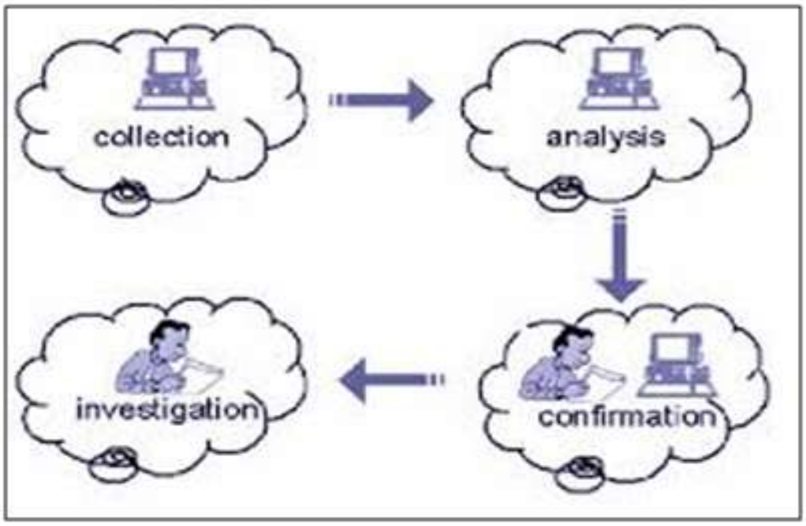

Figure 2: Culwin and Lancaster define a four stage process for detecting plagiarism

Collection: It may be defined as the process of electronically collecting and pre-processing student submissions into a suitable format.
Analysis: It is defined as "where the submissions are compared with each other andwith documents obtained from the Web and the list of those submissions, or pairs,that require further investigation is produced."
Verification (confirmation): It is required to ensure that those pairs reported asbeing suspicious are worth investigating with a view to possible disciplinary action(this is a task normally undertaken by humans, since value judgements may beinvolved).
Investigation: This is the final stage. It will determine the extent of the alleged misconductand will "also involve the process of deciding culpability and possible penalties."

Technologies of Plagiarism Detection:

1. **Text-based plagiarism detection** technology treats the source code as pure text, divides the text by line and compares the code line by line to detect the plagiarism between the files. The main algorithm used in this technology is the Longest Common Subsequence (LCS) algorithm. But, the algorithm has obvious limitations. Only if all strings are the same, the source code can be considered to be similar.
2. **Token-based technology** is the improvement of text-based technology, some famous detecting tools, such as CP-Mine, CCFinder, Winnowing, and JPlag are all token-based. They all cannot detect the modification of renaming, reordering, and inserting null strings.
3. **Tree-based plagiarism** detection is a new technology compared with token-based andtext-based methods, it detects plagiarism from the structureof the syntax tree and can detect those plagiarism cases cannot detected by token-based and text-based technology [3].
4. **Style-based plagiarism detection** is one of the new technologies, it mainly includes two steps. Firstly, the main features representing a coding style are extracted. To determine the plagiarized codes each code is compared to all other codes submitted to the system and also it is compared to its owner's style. Secondly, the extracted features are used in three different modules to detect the plagiarized codes and to determine the giver and takers of the codes.

Tools used for text based plagiarism

Detection tools are present that operate on free text and also that finds similarity in spreadsheets, diagrams, scientific experiments, music or any other non-textual corpora. Tools can be divided in three based on the type of corpus: tools that operate only intra-corpally (where the source and copy documents are both within a corpus), tools that operate only extra-corpally (where the copy is inside the corpus and the source outside) and tools that operate both – intra- and extra-corpally. Most contemporary detection systems adopt a lexical-structural approach to identify these transformations: source programs are tokenized, and profiles are created and compared. While some academic institutions have developed their own in-house detections systems, such as Big Brother, there are also services available through a Web interface. The main players in this field are SIM, YAP, MOSS (Measure of Software Similarity) and JPLAG [4].

1. **SIM (Software Similarity Tester):** The SIM system tokenizes source programs and compares strings using pattern- matching algorithms based on work from the human genome project.
2. **YAP (Yet Another Plague):** The YAP approach also tokenizes source programs and retains only those tokens which are concerned with the structure of the program. This is based on a lexicon which is created specifically for each programming language. The output is a numeric profile which computes the closeness between two programs. This closeness between programs is partly a function of the programming language chosen and the type of ask undertaken (for instance, the COBOL programming language is by nature a highly structured and verbose language, and can lead to very similar programs, likewise with Visual Basic).
3. **MOSS (Measure of Software Similarity):** MOSS can be applied to a range of programming languages. Registered instructors can submit batches of programs to the MOSS server, and results are returned to a website. Little information is available on how the tool works (presumably because if this were known, it would be possible to evade detection), but it is based on the syntax or structure of a program, rather than the algorithms which drive the program The MOSS database stores an internal representation of programs, and then looks for similarities between them.
4. **JPLAG:** JPLAG compares submitted programs in pairs, and is based on the assumption that plagiarists may vary the names of variables or classes, but they are least likely to change the control structure of a program. It has a very powerful graphical interface. The performance of both MOSS and JPLAG were evaluated in the JISC (Joint Information Services Committee) report. The survey concluded that JPLAG was easier to use but supported fewer languages than MOSS and could not deal with programs which do not parse. As students often submit files which do not parse, such a limitation would mean that many files would not be under consideration. Neither JPLAG nor MOSS is easy to use:
5. **Turnitin:** Corpus-based plagiarism detection software takes as input a suspect document and an archived corpus of authenticated documents, compares the suspect document to the corpus, and outputs passages that the suspect document shares with the corpus and a measure of the likelihood that the author plagiarized material. At a more operational level, a corpus-based protocol is implemented in several ways. TurnItIn.com, the most high profile company in the field, employs document source analysis to generate digital fingerprints of documents, those submitted for authentication, those in the archive, and those available through ProQuest.
6. **Glatt:** Glatt Plagiarism Services exemplifies an interrogative approach to plagiarism detection. Work suspected of being plagiarized is given to the Glatt Plagiarism Screening Program, which is free standing, non-Web-based software. The program replaces every fifth word of the suspected paper with a standard size blank, and the student is then prompted to supply the missing words. The number of correct responses, the amount of time intervening between responses, and various other factors are considered in calculating a plagiarism probability index.
7. **PlagScan:**It is a SaaS (available online and on-premises) plagiarism detection software, used by academic institutions and businesses. PlagScan serves teachers and professors to identify plagiarism and educate students on the appropriate usage of sources in academic works as well as protecting copyrights of texts.The software was launched in 2009 by Markus Goldbach and Johannes Knabe. PlagScan serves schools and universities worldwide, with a strong presence in Germany, Austria and Switzerland. By 2015, PlagScan served more than 1000 organizations worldwide- including the University of Jordan, the Fortis College Largo and the Free University of Berlin. In Austria PlagScan is used at more than 300 high schools to check the final thesis (Vorwissenschaftliche Arbeit) of pupils for plagiarism [5].

The core two-step algorithm was developed in 2008 and has been refined on a regular basis ever since. The software recognizes plagiarism as soon as three successive words attune to a different source. PlagScan's database constantly grows. Users can decide optionally if they want to include their submissions to an internal archive to compare future submissions with.

8. **FTIP:** Image plagiarism is a process, when an image or its part is copied and used without any reference to its source. This paper is to present a general method of searching for images or their cropped parts in huge databases. This method can be used as a core of an image plagiarism system.The proposed searching method is based on the techniqueof F-transform and specifically, the F-transform of a higherdegree,

(Fs-transform, s ≥ 0)

The main idea is to represent the source image and the images in the database by discrete functions and apply the F1-transform to these functions in order to obtain their simplified representations in the form of matrices of F1-transform components. Searching of the source image in the database is then based on a comparison of those F1-transform components by computing distances. In this paper results are only observed for whole copied image or cropped image but not for modified image [6].

III. Proposed System

In the proposed system, the input given by user is forwarded to the converter .Converter converts the input in dictionary accessible format. The Converter forwards the data to comparator which compares input & data stored in dictionary The Comparator searches first data in dictionary.If data doesn't exist in dictionary then it is searched on internet.Data found on internet is updated in dictionary, if data also not found on internet then user given data is accepted as new data and updated in dictionary. The extracted data is given to the comparator.According to the comparison data analysis is done using Data analysis tool.At the end output is shown to the user through report, analytical graphs and rating of given document.

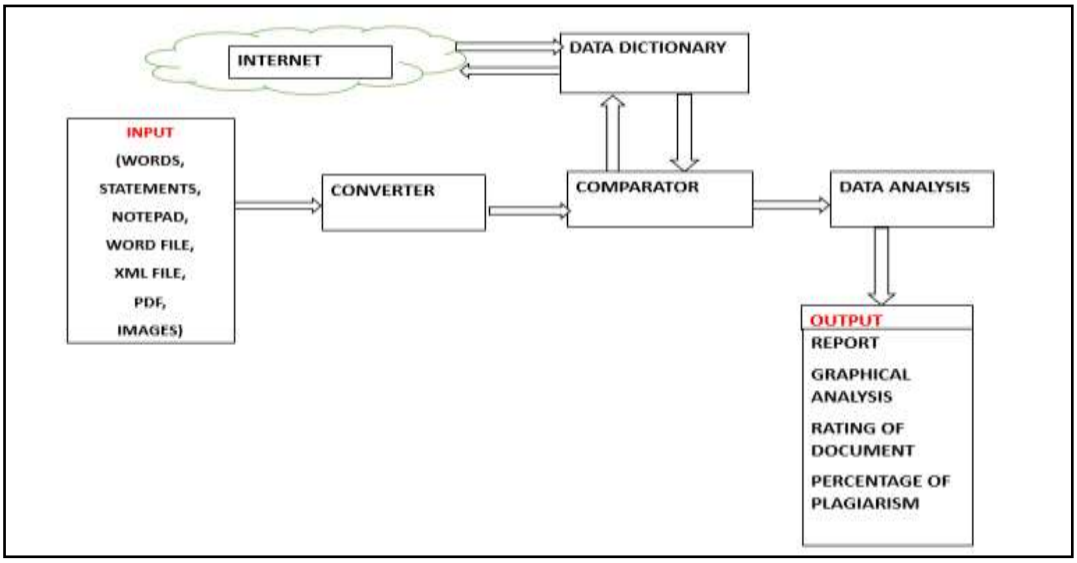

Figure 2: Proposed System

IV. Conclusion

In this technological age, a plagiarism checker is essential for protecting your written work. A plagiarism checker benefits teachers, students, website owners, and anyone else interested in protecting their writing. Our system guarantees that anything you write can be thoroughly checked by our plagiarism software to ensure that your texts are unique.

In this paper, we have described a plagiarism detection system that allows plagiarized documents to be detected. This system is more flexible compared to most of the systems currently present. It allows maximum data to be parsed at a time and process the same using Hadoop. The amount of data analysed is quite large. The percentage wise plagiarism analysis of user data is showed using visualization tools like highcharts, amcharts, etc.

References

[1] http://techterms.com/definition/apache

[2] https://en.wikipedia.org/wiki

[3] "An AST-Based Code Plagiarism Detection Algorithm", Jingling Zhao1,2, Kunfeng Xia 1,2, Yilun Fu 3, Baojiang Cui1, 2015 10th International Conference on Broadband and Wireless Computing, Communication and Applications

[4] https://www.digitalgyd.com/top-20-best-online-plagiarism-checker-tools-free/

[5] http://www.plagscan.com/technology

[6] "FTIP: a Tool for an Image Plagiarism Detection", Petr Hurtik, Petra Hodakova University of Ostrava, Centre of Excellence IT4Innovations, Institute for Research and Applications of Fuzzy Modelling, 2015 Seventh International Conference of Soft Computing and Pattern Recognition (SoCPaR 2015)

[7] "The method for detecting plagiarism in a collection of documents", Natalya Shakhovska, Iryna Shvorob. "Computer Science & Information Technologies" (Csit'2015), 14-17 September 2015, Lviv, Ukraine

[8] http://lucene.apache.org/solr/

[9] https://lucene.apache.org/

[10] http://www.plagscan.com/technology

[11] http://dspace.cusat.ac.in/jspui/bitstream/123456789/3618/1/PDT.pdf

[12] http://ceur-ws.org/Vol-706/poster22.pdf

[13] https://studentnet.cs.manchester.ac.uk/resources/library/thesis_abstracts/MSc08/Abstracts/CornicPierre-fulltext.pdf

[14] http://www.sciencedirect.com/science/article/pii/S1877050915002070

[15] https://github.com/nordicway/moji

[16] http://stackoverflow.com/questions/15194877/plagiarism-detector

[17] https://sourceforge.net/projects/antiplagiarismc/?source=directory
[18] http://www.diva-portal.org/smash/get/diva2:428025/fulltext01.pdf
[19] http://theory.stanford.edu/~aiken/publications/papers/sigmod03.pdf
[20] http://www.prestosoft.com/ps.asp?page=edp_examdiff
[21] http://lucene.apache.org/solr/resources.html