CONTENTS

1. SCOPE OF THE PROJECT
   1. Abstract
   2. Problem Statement
   3. Introduction
2. REQUIREMENTS FOR THE PROJECT
   1. Functional Requirements
   2. Non-Functional Requirements
3. LITERATURE SURVEY
4. PROPOSED METHODOLOGY
5. DATA SETS AND TOOLS
6. EXPERIMENTAL RESULTS AND DISCUSSIONS
7. CONCLUSION
8. REFERENCE LINKS

SCOPE OF THE PROJECT

ABSTRACT:

The wide use of information and communication technologies significantly expanded the possibilities to find out and use information in study process, increased the risk of students to temptation of use some texts or data without referencing the original source which is referred to as Plagiarism. Plagiarism detection for programming language source code represents a topic of growing interest for both the software industry and academia. This project deals with the class of local document fingerprinting algorithms, which seems to capture an essential property of any fingerprinting technique guaranteed to detect copies. We prove a novel lower bound on the performance of any local algorithm

PROBLEM STATEMENT:

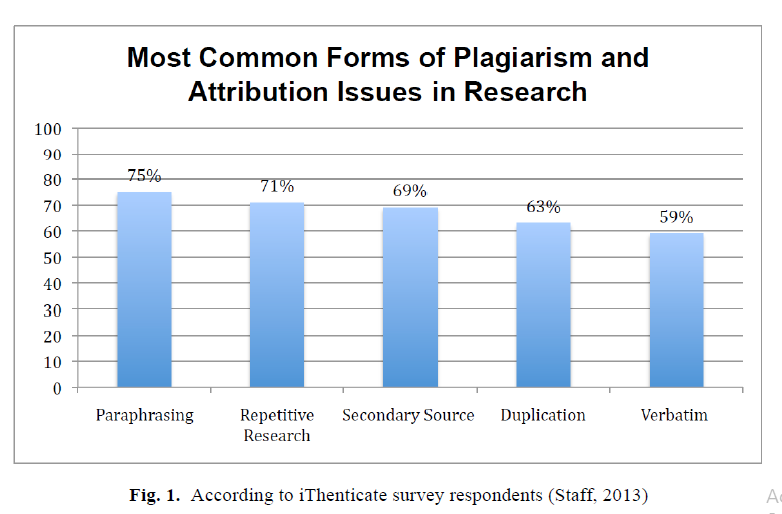
Plagiarism is not new phenomena of academic activity. To detect plagiarism of any form, it is essential to have broad knowledge of its possible forms and classes, and existence of various tools and systems for its detection. Plagiariser of this project detects the percentage of plagiarism taking two input files by using the data mining techniques.

# 

INTRODUCTION:

Detection of source code plagiarism is equally valuable for both academia and industry. “students may plagiarize by copying code from friends, the Web or so called “private tutors”. Most programming courses in universities evaluate students based on the marks of programming assignments. If a programming course consists of a large number of students, it is impractical to check plagiarism by human inspectors. It is known that “A quality plagiarism detector has a

strong impact to law suit prosecution”. Therefore there is a huge demand for accurate source code plagiarism detection systems from both academia and industry. There mentioned two methods for plagiarism detection.



1. *Structured Based Method*: this method considers structural characteristics of documents when developing plagiarism detection algorithms.

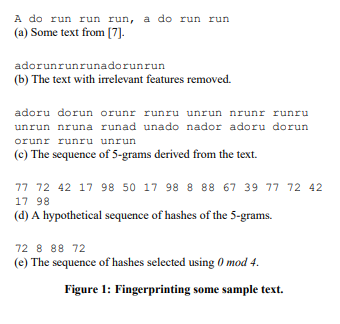
2. *Attribute Counting Method*: this method extracts various measurable features (or metrics) from documents. Extracted metrics use as input for similarity detection algorithms.

Presently most of the source code plagiarism detection algorithms are based on the structured method.Comparing whole document checksums is simple and suffices for reliably detecting exact copies; however, detecting partial copies is subtler. Because of its many potential applications, this second problem has received considerable attention. Most previous techniques for detecting partial copies, make use of the following idea.

A k-gram is a contiguous substring of length k.

Divide a document into k-grams, where k is a parameter chosen by the user.

For example, Figure 1(c) contains all the 5-grams of the string of characters in Figure 1(b). Note that there are almost as many k-grams



Therefore, plagiarism can be classified into various forms. Some are easily detectable and some are complex. Some of the forms are:

**• Coping & pasting**: the type in which a single sentence, a whole paragraph or a complete page of written text is copied without any reference.

• **Re-using existed work:** using again the existing work or already written e-data [4].

• **Manipulating the text:** the type of plagiarism where text is modified and its appearance it changed.

• **Translating the text:** when data is translated from one language to another without giving any reference of the source.

• **Plagiarizing the idea:** one the major form in which someone else’s idea is used without acknowledging the owner.

• **Incorrect citation:** citation of unread sources and without giving acknowledge to the other sources from where the data has been read.

• **Self-plagiarism:** the type in which author uses his own previously done work and presenting as new one with any reference of prior work.

WINNOWING:

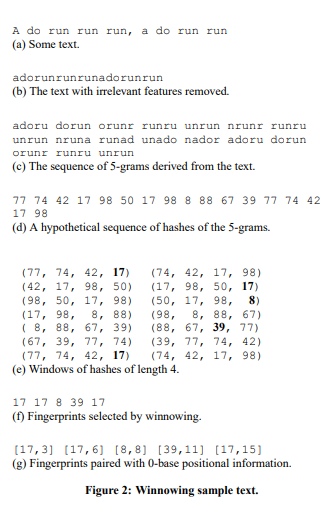
Given a set of documents, we want the find substring matches between them that satisfy two properties:

1. If there is a substring match at least as long as the guarantee threshold, t, then this match is detected, and

2. We do not detect any matches shorter than the noise threshold, k.

The constants t and k ≤ t are chosen by the user. We avoid matching strings below the noise threshold by considering only hashes of k-grams. The larger k is, the more confident we can be that matches between documents are not coincidental. On the other hand, larger values of k also limit the sensitivity to reordering of document contents, as we cannot detect the relocation of any substring of length less than k. Thus, it is important to choose k to be the minimum value that eliminates coincidental matches.

DEFINITION (WINNOWING). In each window select the minimum hash value. If there is more than one hash with the minimum value, select the rightmost occurrence. Now save all selected hashes as the fingerprints of the document. Figure 2(e) gives the windows of length four for the sequence of hashes in Figure 2(d). Each hash that is selected is shown in boldface (but only once, in the window that first selects that hash).



REQUIREMENTS OF THE PROJECT

FUNCTIONAL REQUIREMENTS:

* Detecting the plagiarism between the two input files or strings.
* Displaying the output in terms of similarity.
* The results should be close to the real world data and must be accurate.

NON-FUNCTIONAL REQUIREMENTS:

1. Graphical User Interface for user visibility.
2. Easy to upload the sample code with a single click.
3. Can be extended and used for other languages because of its simplicity.

LITERATURE SURVEY

Plagiarism is defined as the wrongful appropriation or stealing of some other people ideas and make it as own. Stealing or copying of data now a day is becoming very common. Plagiarism detection of copied data originated in 1970’s and common methods of Natural language processing (NLP) for detection of copied data introduced in three different techniques namely Grammar-based method, Semantic-based method and Grammar semantic hybrid method [1].

In **grammar-based method** grammatical structure of the document is maintained and it used a string matching technique for calculating similarity between documents.

**Semantic based method** uses vector space model of the information retrieval technique, and statistical words frequency in document to obtain the vector of the documents, then uses dot product, cousins or other methods to calculate the vectors of two documents. This featured vector is the similarity of the document. This technique isn’t effective as it doesn’t give the source of the plagiarized data.

**Grammar semantic hybrid method** [2] it improves the detection result of these two methods. It is important and effective to highlight or mark the plagiarized text in the documents in parallel to the similarity results. In paper [3] author proposed the Longest Common Consecutive Word algorithm, it considers the whole paragraph as a single unit and tracks the words positions. Then by-word comparison is carried out and common words are obtained, this gives the plagiarized version and similarity between documents.

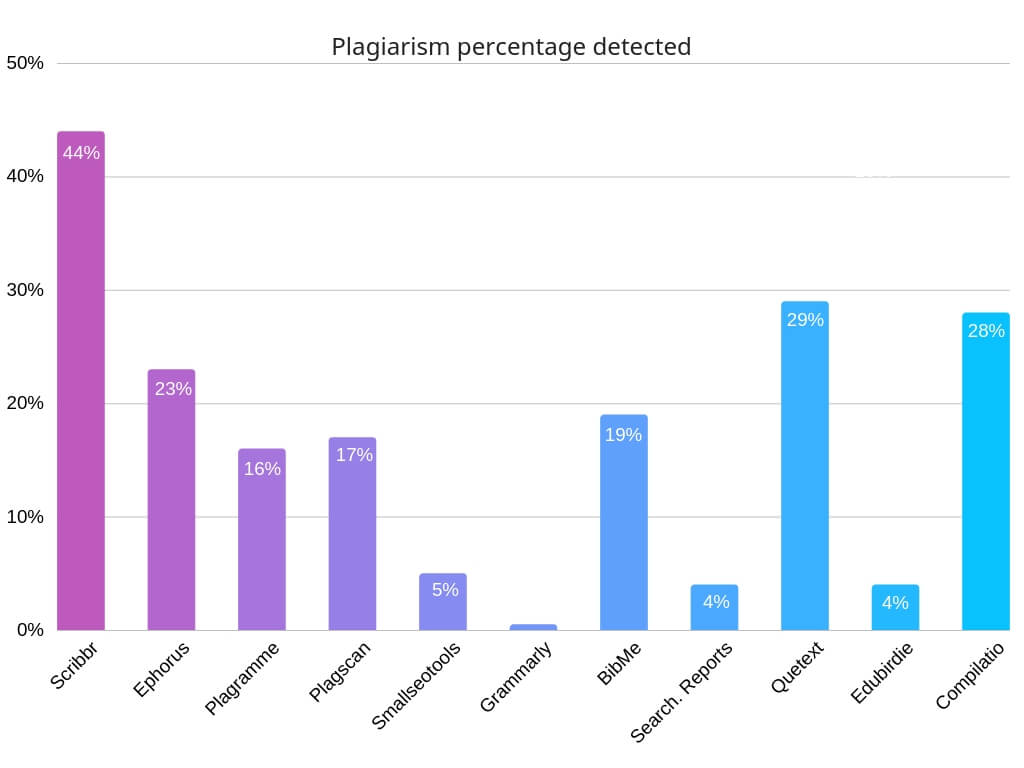


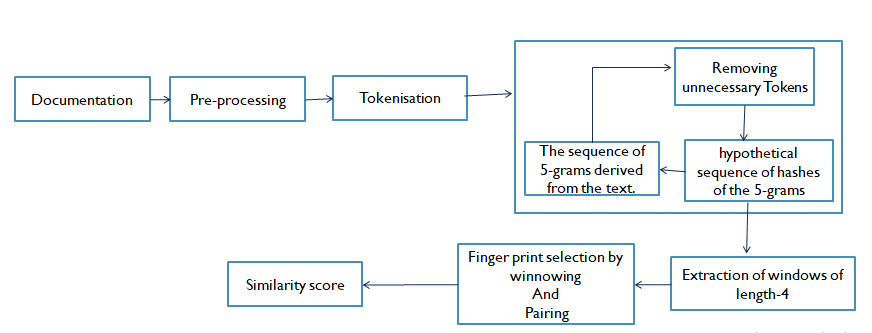
Fig: Plagiarism found in different Content Management services

MDR (Match Detect Reveal) is the method in which the document whose plagiarism is going to be checked is first split into the fixed length strings by maintaining a suffix tree. String matching algorithm is used for comparison, and longest common strings can be found in suffix tree. By this, the similarity index and location in the documents can be obtained. This technique is not efficient because it uses the exact words that match and hence making the unclear plagiarized text version [4].There are different tools which uses web based services and some are standalone

applications.Turnitin, article checker and dupli-checker are most common examplesof web based services, in these tools except turnitin, other provides the free and online text bases plagiarism in limited version whereas turnitin supports both intra and extra corpal detection and is not freely available service. Plagiarism Checker X, Copy-Catch, Plagiarism Detector, WORD-Check and CopyFind are standalone applicationsoftwares. There are many plagiarism detection approaches which can be used by theapplications. Some uses N-gram for improving results in text base. In informationretrieval system precision and recall make much senses in calculating accuracy. But

as compared to N-gram, bi-gram and tri-gram show much better results than n-grams because, tri--gram shows better precision and bi-gram shows better recall. According to authors they assume that tri-gram sequence matching is effective approach [5].

PROPOSED METHODOLOGY

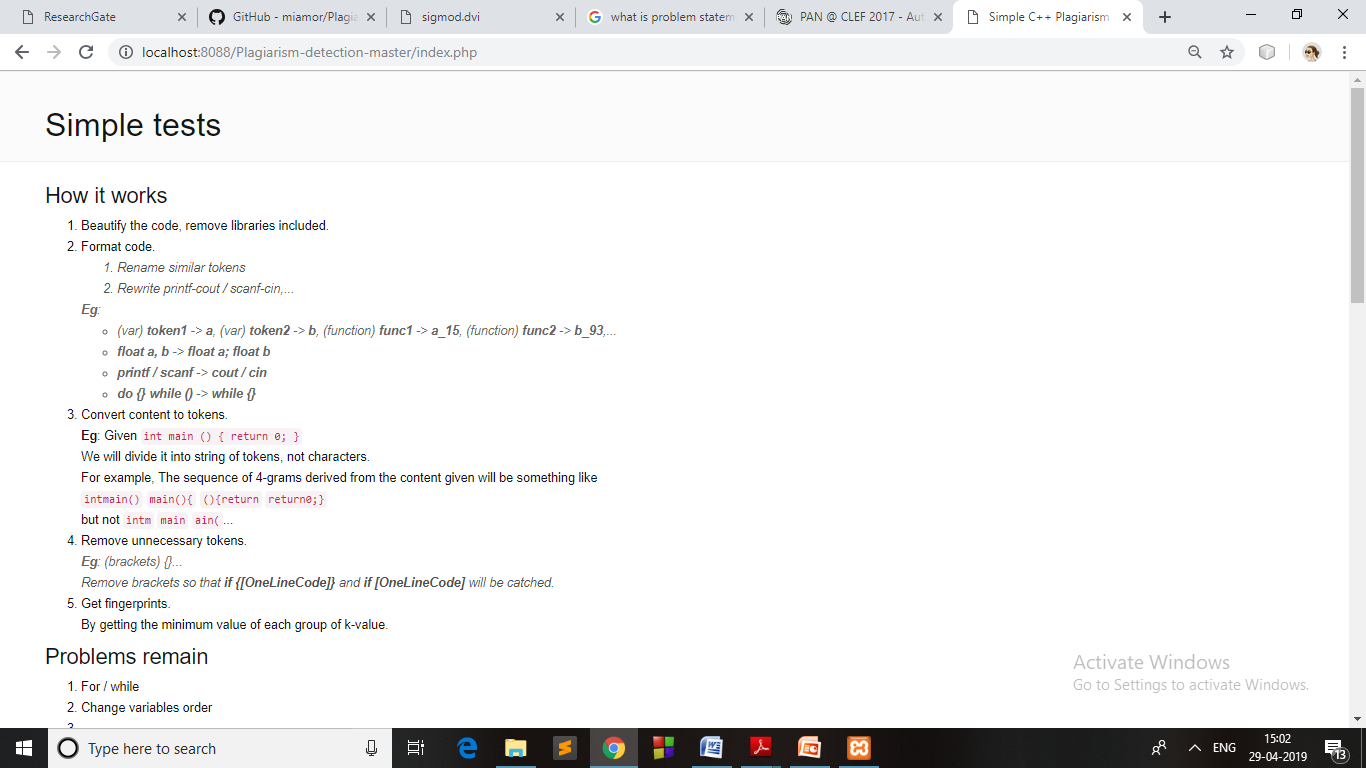


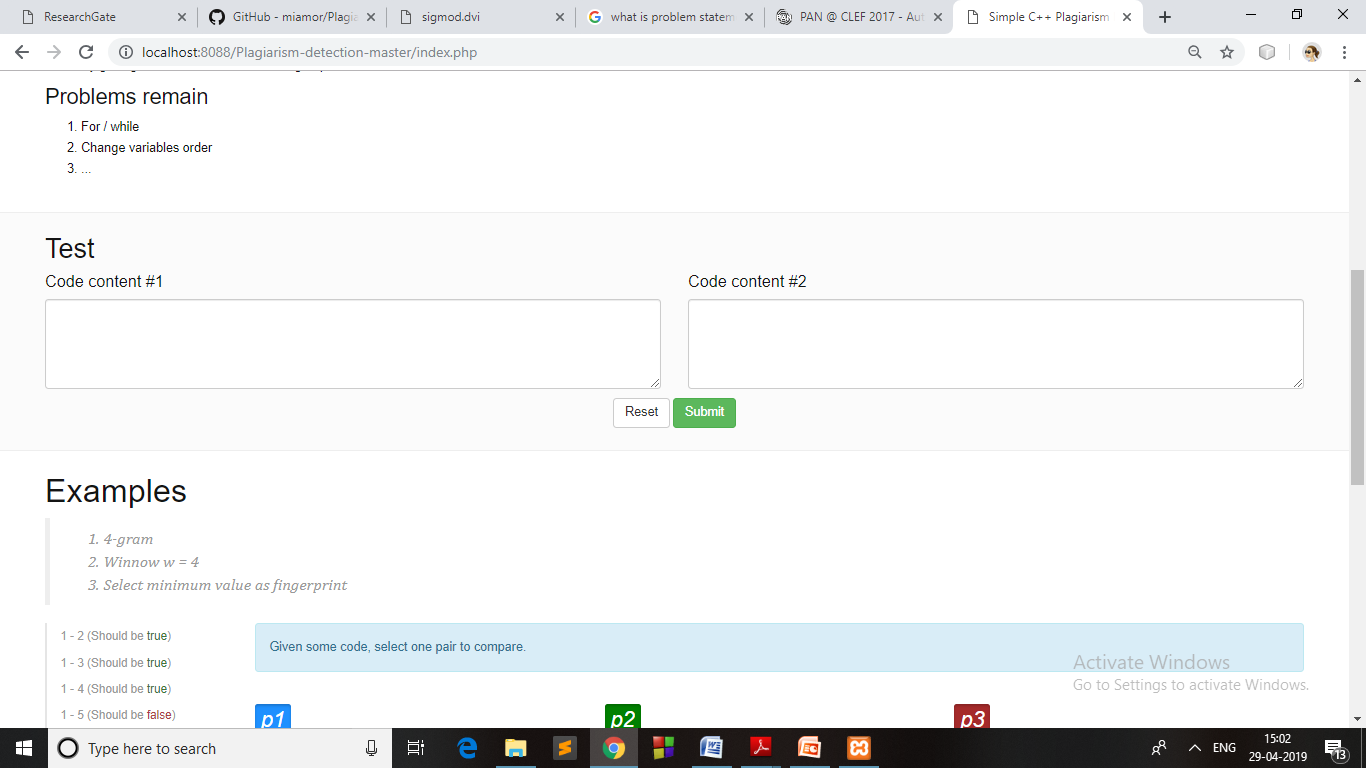
DATASETS AND TOOLS

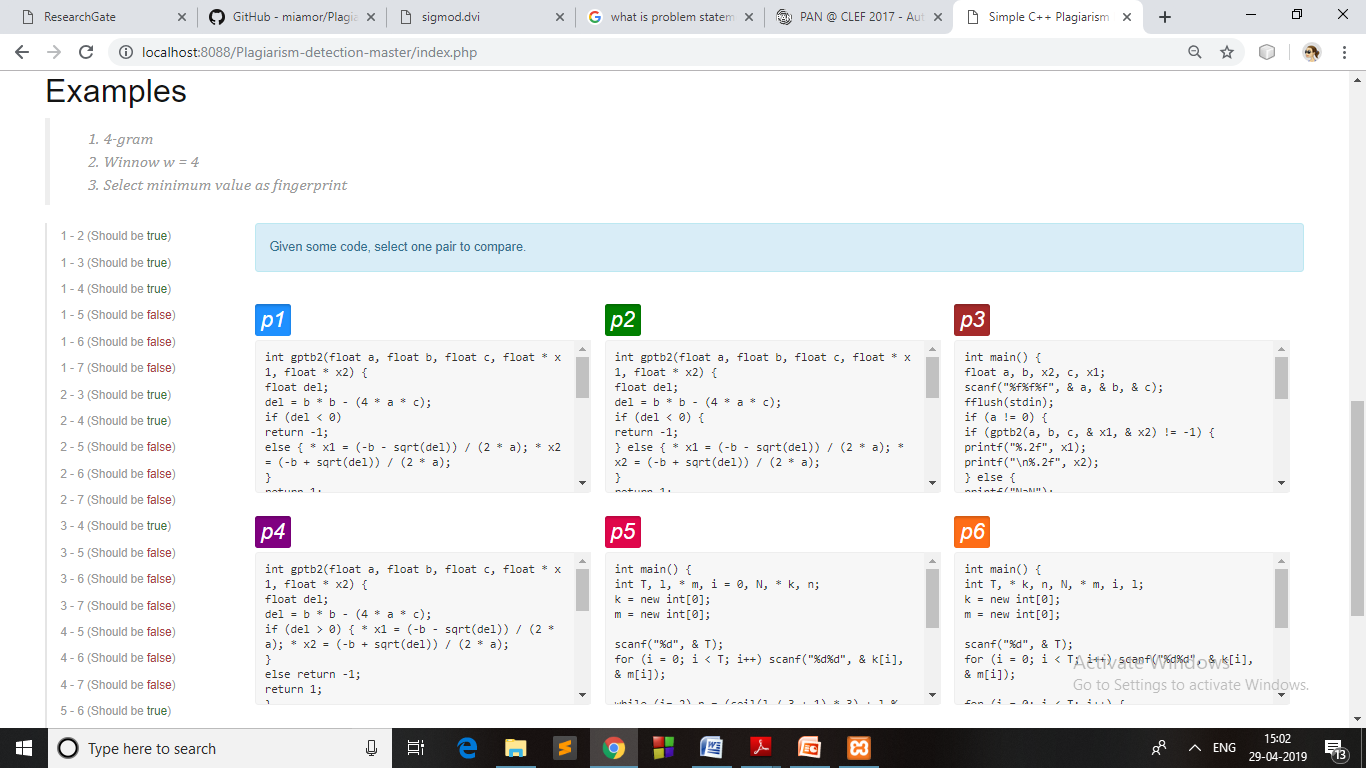
**Dataset:** PAN-10 (2011) for both training and testing

**Languages:** Php and CSS, Java Script

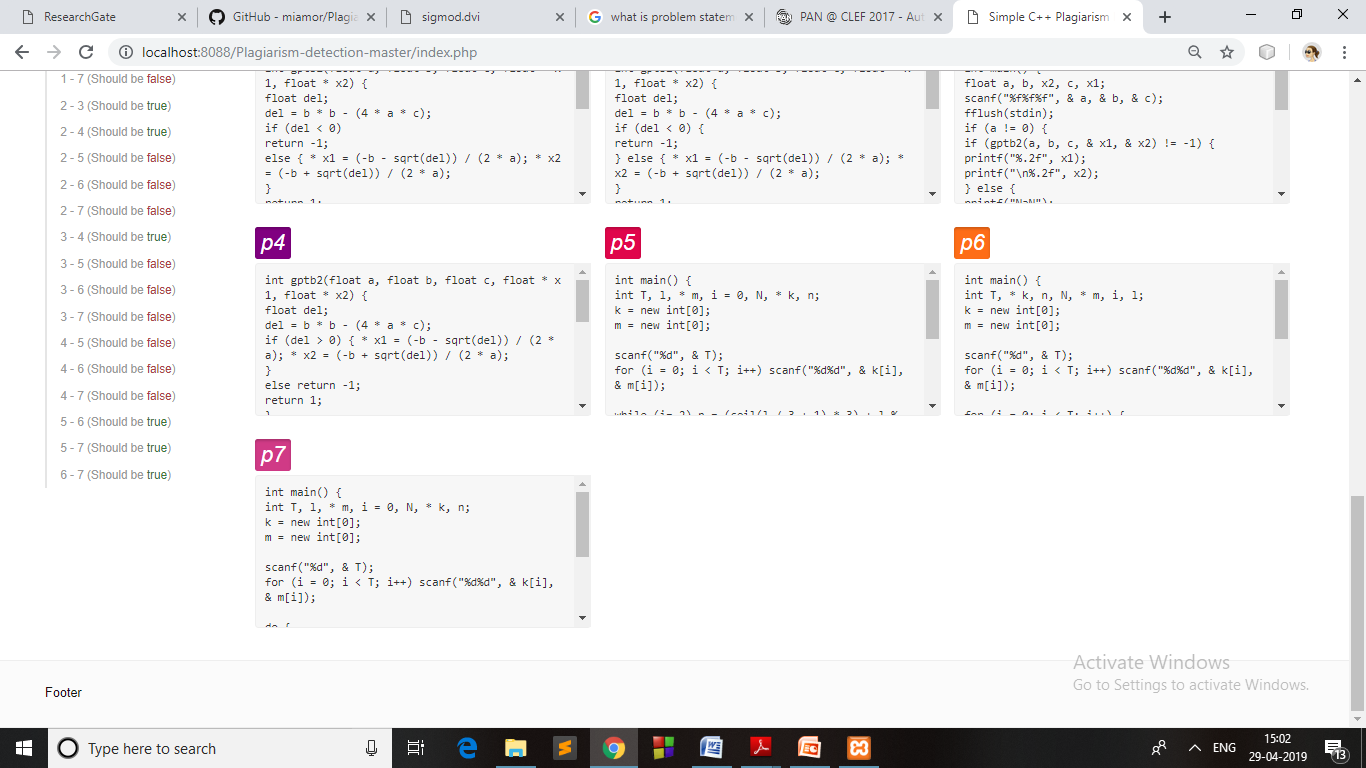
EXPERIMENTAL RESULTS & DISCUSSIONS



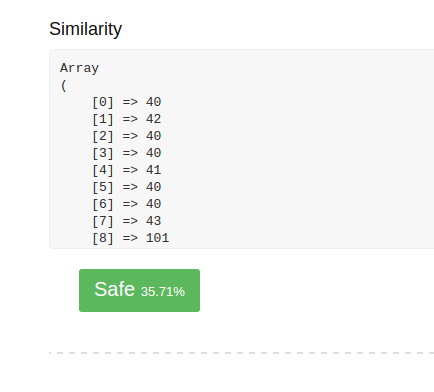


3.

4.



**Result: Similarity in percentage in between the two compared files.**



CONCLUSION

We have presented winnowing, a local document fingerprinting algorithm that is both efficient and guarantees that matches of a certain length are detected. We have also presented a non-trivial lower bound on the complexity of any local document fingerprinting algorithm.

REFERENCES

[1] Jun-Peng, B., & Shen Jun-Yi, L. X.-D.-B. (2003). A Survey on Natural Language Text

Copy Detection. Journal of Software , 14 (10), 1753-1760.

[2] Roig, M. (2011). Avoiding plagiarism, self-plagiarism, and other questionable writing

practices: A guide to ethical writing.

[3] Sediyono, A., & Mahamud, K. (2008 ). Algorithm of the Longest Commonly Consecutive

Word for Plagiarism Detection in Text Based Document. *Digital Information Management*

, 253-259.

[4] Sharma, N., Bajpai, A., & M. R. (2012). Comparison the various clustering algorithms of

weka tools. *International Journal of Emerging Technology and Advanced Engineering , 2*

(5), 73-80.

[5] Staff, A. (2013, 10 15). *iThenticate study ID’s 10 plagiarism formats worthy of attention*.

(American Society of Business Publication Editors) Retrieved 09 21, 2017, from

https://www.asbpe.org/blog/2013/10/15/ithenticate-study-ids-10-plagiarism-formats-worth

y-of-attention/