ISSN No. 0976-5697

Volume 3, No. 3, May-June 2012

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Frequency Based Structural Analysis for Document Plagiarism

Anurag Rai
Department of Computer Science
Krishna Institute of Engineering & Technology,
Ghaziabad-201206, India
anurag.rai13@gmail.com

Pranav Singh
Department of Computer Science
Krishna Institute of Engineering & Technology,
Ghaziabad-201206, India
pranav.singh02@gmail.com

Santosh Kumar Gupta*
Department of Computer Science
Krishna Institute of Engineering & Technology,
Ghaziabad-201206, India
santosh_k25@rediffmail.com

Abstract: The introduction of World Wide Web and the increasing standardization of electronic documents have led to information being easily available on the Internet. This has led to a greater incidence of plagiarism in every aspect of life. Easier availability of research papers, books, technical and non-technical papers, etc. are the easiest source for making plagiarized documents. Therefore it becomes a very daunting task to detect plagiarism, particularly when an attempt is made to disguise the plagiarism, i.e., using different words to put forward the same idea. This paper uses keywords based matching to detect plagiarism. The method is based on the making of the decision tree then performing structural analysis of the complete document followed by comparison of subsection of the document along with the frequency of word.

Keywords: Plagiarism, Keywords Based Matching, Decision Tree, Structural Analysis

I. INTRODUCTION

Plagiarism is most challenging problem in publishing of scientific, engineering and other types of documents. Plagiarism is increased with the easy and abundant availability of information through widespread use of the Internet. Plagiarism is not just direct copy but also rephrasing, rewording, adapting parts, missing references or wrong citations. This way the problem becomes more difficult to be handled properly. Most of the document plagiarism detecting software is not very efficient. Documents could be easily bypassed those software. Most of them employed an exhaustive sentence based comparison technique to detect plagiarism. This approach is not scalable to a large and diverse set of documents.

When a potential plagiarized document is compared against a registered document, an information retrieval technique is applied to preprocess the documents and determine the semantic meaning of the document. If the documents are on different subject, then further comparisons could be avoided.

Plagiarism can be defined as the act of taking or attempting to take or to use (whole or parts) of another person's works, without referencing or citation him as the owner of this work. It may include direct copy and paste, modification or changing some words of the original information from the internet books, magazine, newspaper, research, journal, personal information or ideas.

Plagiarism was originated in 1970’s and then in 1990’s it became more popular among researchers. Actually after the development of the internet, plagiarism has become a big problem among the institute and the university. The originality of the work in any document has come into danger. Plagiarism, as in [9], is the “use or close imitation of the language and thoughts of another author and the representation of them as one’s own original work". Plagiarism detection system is basically detection of plagiarism in different document, assignments or research papers. Plagiarism is the act of taking the writings of another person and passing them off as one’s own. This act is closely related to forgery and piracy practices which generally results in violation of copyright laws as cited as in [10].

The paper is organized as follows: Related work is given in Section 2 followed by Plagiarism and Classification of Plagiarism Detection Systems are discussed in Section 3. Section 4 discusses about Text Based Plagiarism. Proposed method is given in Section 5. An example is given in Section 6. Section 7 is the conclusion.

II. RELATED WORK

Plagiarism detection originated from program code similarity detection in the 1970’s. Text-based copy detection technique basically appeared for the first time in early 1990s. Many attempts have been made in the past to detect plagiarized documents.

In [1], document plagiarism detection software that eliminates most of the unnecessary comparisons has been proposed. The elimination of unnecessary comparison like documents having different subject area, improves the efficiency of the tool. When a potential plagiarized document is compared against a registered document, an information retrieval technique is applied to preprocess the documents and determine the semantic meaning of the document. If the documents are on different subject, then further comparisons could be avoided. For preprocessing, the document is parsed which includes document recognition, keyword extraction and structural characteristic generation. The document is compared in the document comparison module and thus the comparison result is shown.

It has been shown in [2], that the most popular commercial systems use RKR-GST based algorithms. According to this method, it is not necessary for the strings to be properly placed next to each other (contiguous) order. This allows documents to be compared even if some text is deleted or additional text has been added in the suspected document. It is also possible for the algorithm to detect plagiarism in documents which has been made by combining text material from different documents. The algorithm used in this paper can be more effective if we parse the documents to remove trivial words and tokens.

An online system for plagiarism detection has been proposed in [3]. This is a plagiarism detection system which searches the Internet for evidence of plagiarism within a document. This is achieved using the PHP scripting language in conjunction with the Google Internet search engine and various Linux applications. The system is a fully web-based server-side program. The LAMP (Linux Apache MySQL PHP) approach was used for the system. Using LAMP is beneficial as the LAMP components are free and open source. GNU/Linux applications that were used are *w3m* and *ps2ascii*. Google was the online search engine used due to the wide use against other search engines.

A case study of the University of Glamorgan has been used to introduce an institutional journey on electronic plagiarism detection to inform the initial experience of an innovative tool and method which could be further explored in the future research [4]. The working of the system is like, Lecturers create assignment link. Then the students submit assignment to through the submission link in advance of the due date. Turnitin UK then returns the digital receipt and originality report. By looking at the originality report, students could enhance their work. After the due date, lecturers mark all submitted assignments either offline or online.

Some modern plagiarism detection software, like Sherlock, JPlag and MOSS has used tokenization technique to improve detection [5]. Before converting the file into the tokenized form, white spaces and comments were removed. A naïve implementation of this comparison, resulted in complexity $O(f(n)N^2)$, where N is the size & f(n) is the time taken for comparisons between one pair of files of length n. The system proposed by the author is based on an index structure built over the entire file collection. Before the index is built, all the files in the collection are tokenized. A suffix array is used as an index structure. An algorithm is used for finding all files within the collection's index that are similar to a given query file. It tries to find the substrings of the tokenized query file, in the suffix array. Matching substrings are recorded and each match contributes to the similarity score. Depending on the similarity score the judgment is made.

In [6], two phase detection method of plagiarism detection system has been discussed. Using this method the author tries to find out the copied text and their location in the document. In this system, the method is divided mainly in two phases i.e. pre-selecting and locating. In pre-selecting phase, suspicious document is broken down in different fingerprints and to improve efficiency successive fingerprint methods were used on the document. In locating phase, we compare the suspicious document with the source document. This method is called clustering based plagiarism detection method which uses Winnowing's fingerprint extracting algorithm. This method basically focuses on the text being copied and to find out the location of the text which has been copied.

A method based on natural language has been used in [7] to detect plagiarism. The SCAM (Stanford Copy Analysis Mechanism) detects plagiarism by comparing set of words that are common between registered and test document. The analysis of document can be semantic or statistical. For the document representation, Vector Space model is used. For searching of documents, Apache Lucene java library is used and Lucene functionalities are used for searching algorithm. The result is displayed on GUI as a score of comparison. The two fundamental concepts for the evaluation of an information retrieval system are precision and recall. These parameters are responsible for the effectiveness of plagiarism detection. This method is efficient in finding out the exact or partial text plagiarised. For the rewording and paraphrasing, WordNet is being used which increases the effectiveness of the algorithm.

In [8], different plagiarism detection method in order to find out different aspects of plagiarism has been discussed. There are large numbers of method being used for PDS in past years. Here are some of the most prominent methods used these days: Grammar based method, Semantics based methods, Grammar semantics hybrid methods. Many more methods are classified under these main methods. The detection of plagiarism, about 100 % accurate, is a difficult task. Clustering has become the techniques of sorting and summarizing tool both. Semantics based methods for cluster based methods are used for better results. In the coming future, the PDS will focus on supported language, extendibility, presentation of results, usability, exclusion of template code, exclusion of small files, historical comparisons, submission of file-based rating, local or web based and open source.

In [9], optimized pre-processing model has been proposed to detect plagiarism in a large repository. This method uses GDIC (Global DICtionary) for pre-processing. Now two methods are used for inspection and these two methods work at same time. Both of them are based on concept of a common non-stopword chosen pairs. Plagiarism detection time depends on number of pairs used to inspect plagiarism, so in order to get better result search space should be small. GDIC is an efficient new data structure for detecting similar texts. GDIC method is used for reducing the search space and thus the search time can be easily reduced. Using GDIC, new data structures are used to improve the inspection capability for a large number of documents. Using this method, search time decreases by around 20%, computation time becomes less and GDIC requires very less time of the operating system. In the future, pre-processing will be done with simplified and optimized GDIC.

In [10], effective clustering and faster searching approaches is used in plagiarism detection. In this paper singular value decomposition is used for effective clustering of document and neural networks for local matching and comparison. Kohonen maps (Self-organizing maps (SOM)) are used to visualize and compare the results.

III. PLAGIARISM DETECTION

Plagiarism detection is a detection system which is designed to check whether a document is copied from other documents in whole or only a part of it without any references. Plagiarism can be defined as the claiming or implying original authorship of other’s idea or concepts or material, written or creative work, in whole or in part, into one's own without adequate acknowledgement. Besides all this, copying text without any change, changing the order of the original text and replacing the word with their synonyms are also considered as plagiarism. Plagiarism detection technology is broadly used for the protection of intellectual properties, copyrighted documents, search engines, e-libraries and student paper checks.

A. Classification of Plagiarism Detection Systems

Basically, there is no single criterion for performing classification. Most hermetic systems are either universal (that is, can process text documents of any nature) or specially designed to detect plagiarism in source code files. The figure below shows the classification of plagiarism detection systems based on algorithm.

(a) Fingerprint-Based Systems: The main idea of fingerprinting is to create fingerprints for all documents in a collection. Fingerprint is short sequence of bytes that characterizes a longer file. For instance, fingerprints can be obtained by applying any hash function to a file. In plagiarsim detection systems, fingerprints are more advanced than simple hash codes. Nowadays, it is generally believed that attribute counting is inferior to content comparison, since even small modifications can greatly affect fingerprints. As a result, later systems usually do not follow this technique, but there are several recent systems that combine fingerprinting with elements of string matching, for example, MOSS program.

(b) Content Comparison Techniques: These are the building blocks of majority of the present plagiarism detection systems. There are different algorithms aimed at file-file comparison, varying in terms of speed, memory requirements and expected reliability.

(c) String Matching Based Content Comparison: String matching based methods compare files by treating them as strings. It usually does not take into account the hierarchical structure of the computer program, considering it as raw data.

(d) Parse Trees Comparison: A parse tree is an ordered or rooted tree which basically represents the structured form of a strings or expressions according to the formal grammar. Parse trees can be generated for sentences in natural language as well as for the coding logics in programming languages. Natural language texts are divided into sections, subsections, paragraphs and sentences, while source code files contain classes, functions, logic blocks and control structures. Though this approach seems to be the most advanced, little research in this area has been carried out so far. For example, it is still unknown how such a complex analysis of input files influences the final results, that is it is undiscovered whether parse trees One approach to address the issue of plagiarism is to provide copy detection systems to which legal original documents are registered and copies are detected.

IV. TEXT BASED PLAGIARISM & ITS DETECTION

Copying text from another source, even partially, without giving credit to the person who actually wrote it and not enclosing the text copied, in quotation marks is text based plagiarism.

A. Some Important Guidelines:

- Any text copied from another source must be enclosed in quotation marks.
- We must always mention every source that we use in our writing; whether we use the verbatim, summarize, or enclose it within quotation marks.
- Whether we are copying directly or taking the basic idea from a source, we must always identify the source of our information.
- In order to make significant changes to the original text that results in a proper paraphrase, the author must have a thorough understanding of the ideas and terminology being used.
- It is the ethical responsibility of the writer towards the readers, and to the authors from whom s/he is borrowing, to respect others’ ideas, to give credit those from whom he borrow, and whenever possible, to use one’s own words when paraphrasing.
- When one is not sure as to whether a concept or fact is common knowledge, to be on the safer side, provide a citation.

Fig. 4.1 shows a classification of methods for computer-assisted plagiarism detection. The techniques are characterized by the type of similarity assessment they use. Global similarity assessments use features taken from larger parts of the text or the document as a whole for similarity computation, while local methods take confined text segments as input [11].

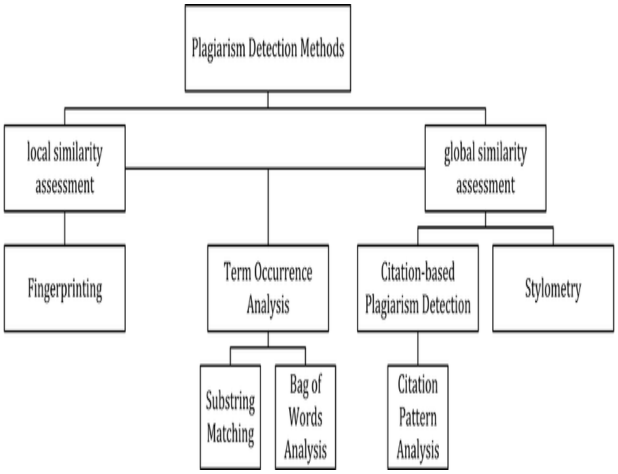

Plagiarism Detection Methods
local similarity assessment
global similarity assessment
Fingerprinting
Term Occurrence Analysis
Citation-based Plagiarism Detection
Stylometry
Substring Matching
Bag of Words Analysis
Citation Pattern Analysis

Figure: 4.1 (Wikipeida)

Fingerprinting is widely used in computer-assisted plagiarism detection. The procedure forms representative digests of documents by selecting a set of multiple substrings (n-grams) from them. These sets represent the fingerprints and their elements are called minutiae [11]. To check for plagiarism, the fingerprint for the suspicious document is computed and minutiae queried with a pre computed index of fingerprints for all documents of a reference storage. The document will be said to be plagiarised if minutiae is found to be matching, exceeding a chosen similarity threshold, with those of other documents.

Checking documents for plagiarism represents a classical string matching problem known from other areas of computer science. A significant amount of work has been done in this field to tackle this task, of which external CaPD is one to which some have been adapted to. Computation and storage of comparable representations for all documents in the reference storage is required for checking a suspicious document, checking being done pair-wise. Nonetheless, matching of substring remains a computationally expensive task, which makes it a non-preferred solution for checking large collection of documents.

Plagiarism detection based on citations is a plagiarism detection approach, assisted by computer, and is designed for usage with academic documents, since it does not depend on the text itself, but on citation and reference information. It identifies similar patterns in the sequence of citation of two academic works. Citation patterns represent subsequences containing citations common to both the documents being compared. Similar order and propinquity of citations within the text is one of the main criteria for identifying citation patterns.

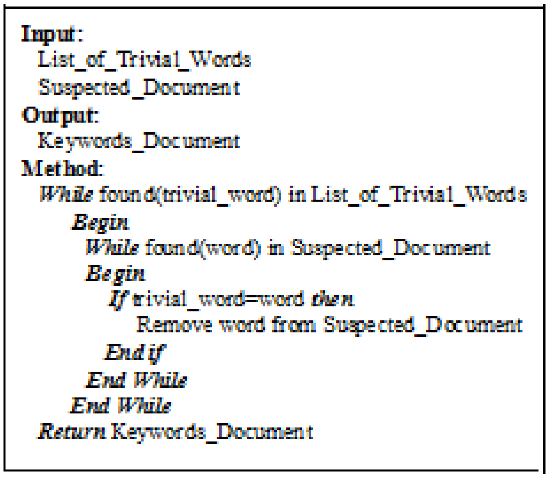
Stylometry subsumes statistical methods for quantifying an author's unique writing style and is mainly used for authorship attribution or intrinsic CaPD. By developing the stylometric models for different text segments and then comparing, the portion that is stylistically different from others, hence potentially plagiarized, can be detected.

V. PROPOSED METHOD

In the methodology proposed for implementing the document plagiarism detection system, the main theme for checking plagiarism is the comparison of document tree of the suspected document and the set of base documents stored in the database. It is important to mention here that the comparison will be done against a pre-stored set of documents and their pre-constructed document tree, in the database. In the methodology proposed for implementing the document plagiarism detection system, the main theme for checking plagiarism is the comparison of document tree of the suspected document and the set of base documents stored in the database. It is important to mention here that the comparison will be done against a pre-stored set of documents and their pre-constructed document tree, in the database.

The suspected document is scanned and entered into the system. The input document is recognized and further processed to prepare it suitably for comparison with a pre-stored set of documents.

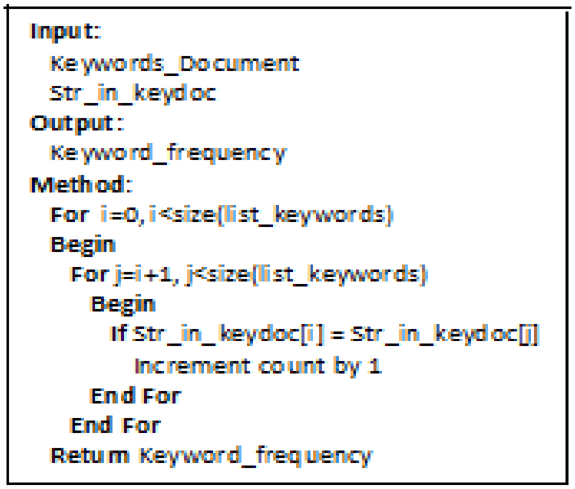
A document containing the list of trivial words like conjunctions, articles, adverbs, etc. is maintained separately. The purpose of maintaining this document is to scan the input document for these non-important words and remove them because anyhow these words will not be useful for detecting plagiarism. Therefore, the input document is scanned for these trivial words and subsequently deleted, and the document with the trivial words deleted is saved for further processing. The procedure to remove trivial words from suspected documents is described in Fig. 5.1.



```
Input:
  List_of_Trivial_Words
  Suspected_Document
Output:
  Keywords_Document
Method:
  While found(trivial_word) in List_of_Trivial_Words
      Begin
        While found(word) in Suspected_Document
        Begin
          If trivial_word=word then
              Remove word from Suspected_Document
          End if
        End While
      End While
  Return Keywords_Document
```

Figure: 5.1

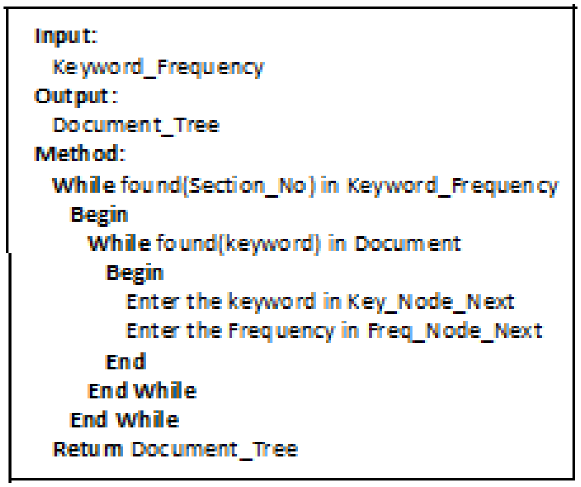
The document saved at the previous step is processed to extract the words, which are now called as keywords for the document, from it. The frequency of each extracted keyword is counted and this information is written into a new document. This process is exhaustive and hence, done for each keyword. The pseudo code for counting the frequency of each keyword in the document is given in Fig. 5.2.



```
Input:
  Keywords_Document
  Str_in_keydoc
Output:
  Keyword_frequency
Method:
  For i=0, i<size(list_keywords)
  Begin
    For j=i+1, j<size(list_keywords)
      Begin
        If Str_in_keydoc[i] = Str_in_keydoc[j]
          Increment count by 1
      End For
    End For
  Return Keyword_frequency
```

Figure: 5.2

The next step is a very important step and forms the backbone of the system. Now a document tree is constructed from all the information derived up to the previous step. The pseudo code for constructing document tree is given in Fig. 5.3. For a given document which is divided into sections, a node of the tree stores the information regarding a particular section. For example, in the Fig. 5.4, the rightmost child of the root node stores the information about section 1 in the suspected document. Now the left child of this node will store information regarding section 1.1, and so on. The root node acts as the parent of the whole document and stores all the information about the document. The information which the nodes are storing is nothing but the keywords with their frequency. The importance of this document tree is that this document tree is what will be compared for checking the plagiarism between the documents. This will be more evident in the next step.



```
Input:
  Keyword_Frequency
Output:
  Document_Tree
Method:
  While found(Section_No) in Keyword_Frequency
    Begin
      While found(keyword) in Document
        Begin
          Enter the keyword in Key_Node_Next
          Enter the Frequency in Freq_Node_Next
        End
      End While
    End While
  Return Document_Tree
```

Figure: 5.3

In this step, the comparison module is invoked. In this module, the document tree of the suspected document is compared against the document tree of the pre-stored set of documents in the database.

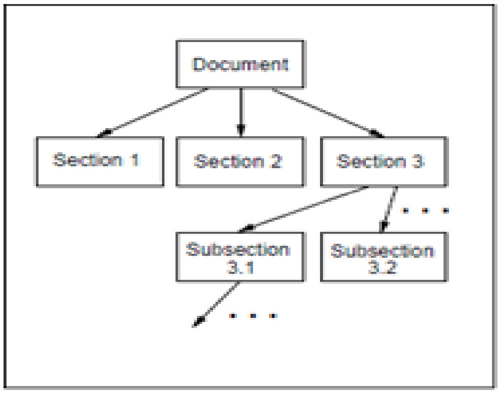


Figure: 5.4

Initially, the root nodes of the two documents will be compared. If there will be matching of a minimum number of the keywords with 50 percent of the frequency also matching, then the comparison will be done further at the next level of the tree, taking the child nodes now as the root nodes, to detect the exact point where plagiarism is done. This is depicted through Fig. 5.5. In the next step both suspected and original documents are to be compared.

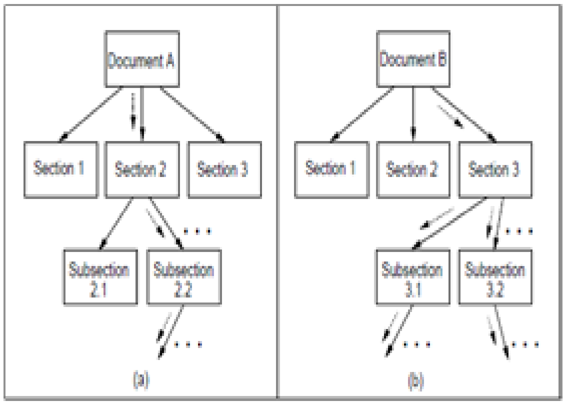
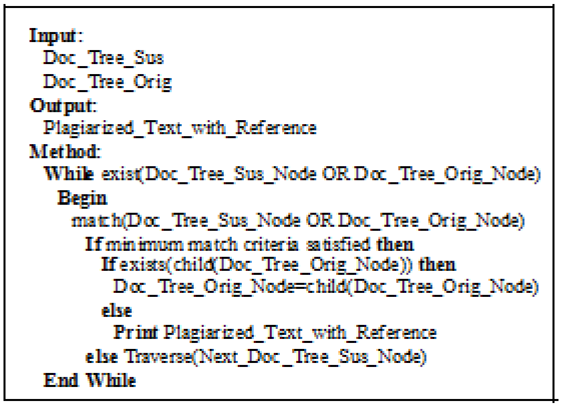


Figure: 5.5

The pseudo code for the comparison of suspected document with the documents in the database is given in Fig. 5.6. This returns the result as plagiarized text with reference, in the original document, of each line or paragraph in the suspected document.



```
Input:
  Doc_Tree_Sus
  Doc_Tree_Orig
Output:
  Plagiarized_Text_with_Reference
Method:
  While exist(Doc_Tree_Sus_Node OR Doc_Tree_Orig_Node)
    Begin
      match(Doc_Tree_Sus_Node OR Doc_Tree_Orig_Node)
        If minimum match criteria satisfied then
          If exists(child(Doc_Tree_Orig_Node)) then
            Doc_Tree_Orig_Node=child(Doc_Tree_Orig_Node)
          else
            Print Plagiarized_Text_with_Reference
        else Traverse(Next_Doc_Tree_Sus_Node)
  End While
```

Figure: 5.6

If the document is found to be original, then the suitable message is printed, and because the suspected document has originality, therefore it is also registered in the database for comparison with suspected document in the future.

The flow chart for the overall process is given in Fig. 5.7 and it clearly depicts the methodology used in a very lucid manner.

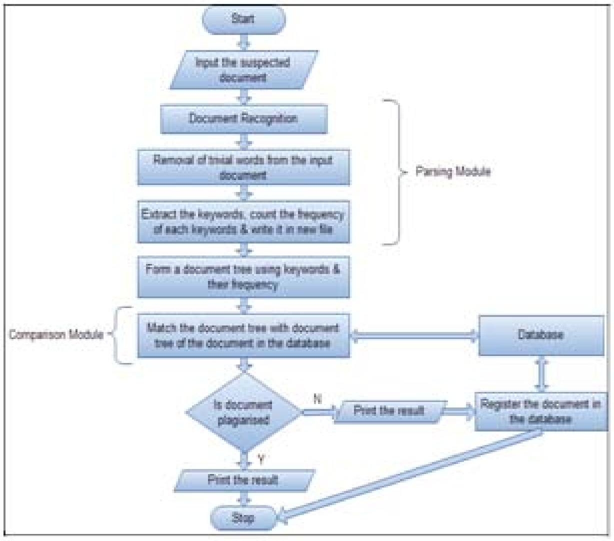

Start
Input the suspected document
Document Recognition
Removal of trivial words from the input document
Extract the keywords, count the frequency of each keywords & write it in new file
Parsing Module
Form a document tree using keywords & their frequency
Comparison Module
Match the document tree with document tree of the document in the database
Database
Is document plagiarised
N
Print the result
Register the document in the database
Y
Print the result
Stop

Figure: 5.7

VI. AN EXAMPLE

Let us explain the methodology proposed with an example. Given below are contents of two documents, first is the original one and the second is the suspected document.

The original document is given in Fig.6.1

1. **Renaissance**
The word 'Renaissance' is a French term first coined in the 19th century to describe the intellectual and artistic revival, inspired by a renewed study of Classical literature and art, which began in Italy in the early 14th century and reached its culmination in the early 16th century, having spread in the meantime to other parts of Europe.
1.1 **Brief History**
Until the 20th century the generally accepted model for the development of the artistic Renaissance was that constructed by Vasari, writing in 1550. He gave to Giotto the credit for the rebirth of art after centuries of barbarism and structured his chronological model.
2. **European Sovereign Debt Crisis**
From late 2009, fears of a sovereign debt crisis developed among investors concerning rising government debt levels across the globe together with a wave of downgrading of government debt of certain European states. Concerns intensified early 2010 and thereafter making it difficult or impossible for Greece, Ireland and Portugal to re-finance their debts.
2.1 **Causes**
The European sovereign debt crisis has been created by a combination of complex factors such as: the globalization of finance; easy credit conditions during the 2002-2008 period that encouraged high-risk lending and borrowing practices; international trade imbalances; real-estate bubbles that have since burst; slow growth economic conditions after 2008."

Figure. 6.1

The suspected document looks as in Fig. 6.2.

1. **Euro Zone Crisis**
From late 2009, fears of a sovereign debt crisis developed among investors concerning rising government debt levels across the globe together with a wave of downgrading of government debt of certain European states. Concerns intensified early 2010 and thereafter making it difficult or impossible for Greece, Ireland and Portugal to re-finance their debt."

Figure. 6.2

In the first step, the trivial words will be removed from the suspected document and the new document will look as in Fig 6.3.

1.0 Euro Zone Crisis
late 2009, fears sovereign debt crisis developed investors concerning rising government debt levels globe wave downgrading government debt European states. Concerns intensified 2010 difficult impossible Greece, Ireland Portugal re-finance debt.

Figure. 6.3

In the next step, the frequency of each keyword will be counted and written back to the document. The document will look as in Fig. 6.4.

late: 1, 2009: 1, fears: 1, sovereign: 1, debt: 4, crisis: 1, developed: 1, investors: 1, concerning: 1, rising: 1, government: 2, levels: 1, globe: 1, wave: 1, downgrading: 1, European: 1, states: 1, concerns: 1, intensified: 1, 2010: 1, difficult: 1, impossible: 1, Greece: 1, Ireland: 1, Portugal: 1, re-finance: 1.

Figure. 6.4

Now a document tree is constructed for the suspected document. Since there is only one section in the suspected document, therefore there will only be a single node (root node) in the document. The document tree looks as shown below in Fig. 6.5.

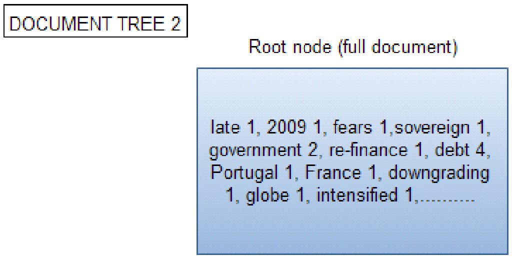

DOCUMENT TREE 2
Root node (full document)
late 1, 2009 1, fears 1,sovereign 1, government 2, re-finance 1, debt 4, Portugal 1, France 1, downgrading 1, globe 1, intensified 1,..........

Figure. 6.5

The document tree for the original document is retrieved from the database as in Fig. 6.6.

Now when we compare these two document trees, we get a certain degree of matching at the root node. So, further comparison is done at the next level of the tree. The node for section 1 in the original document does not match with the node of the suspected document, therefore node 2 of the original document is compared with the node of the suspected document. Now matching is found and further node for section 2.1 in the original document is used for comparison. However, here no matching is detected. Therefore, an algorithm for matching string is used to match the text in section 2 of the original document and the suspected document. On doing so, plagiarism is detected and accordingly the result is printed.

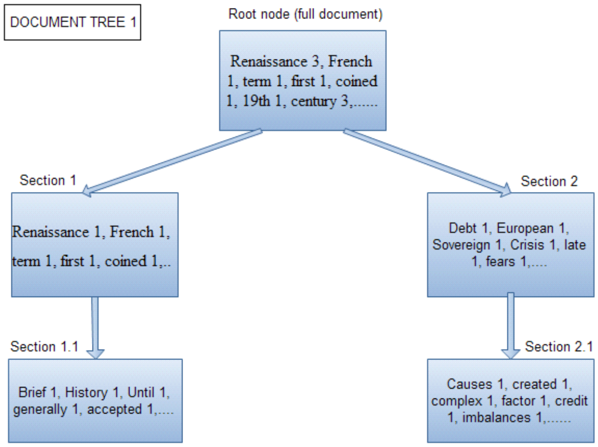

DOCUMENT TREE 1
Root node (full document)
Renaissance 3, French 1, term 1, first 1, coined 1, 19th 1, century 3,......
Section 1
Renaissance 1, French 1, term 1, first 1, coined 1,..
Section 2
Debt 1, European 1, Sovereign 1, Crisis 1, late 1, fears 1,....
Section 1.1
Brief 1, History 1, Until 1, generally 1, accepted 1,....
Section 2.1
Causes 1, created 1, complex 1, factor 1, credit 1, imbalances 1,......

Figure: 6.6

VII. CONCLUSION

The proposed method for detecting text plagiarism would be an efficient method detecting plagiarism due to the elimination of unnecessary comparisons between documents from two different subject areas such as a original document is on chemistry and the suspected document is of computer then there would not be any matching of the document tree after matching the root nodes thereby reducing the time of plagiarism detection hence increasing the efficiency of detection system. The proposed algorithm has been implemented and tested on different set of documents and it showed its correctness.

VIII. REFERENCES

[1]. Si A., Leong H.V.and Lau R.W.H., “CHECK: A Document Plagiarism Detection System” in “ACM Symposium for Applied Computing”, Feb. 1997, pp. 70-77.

[2]. Noynaert J.E.,“Plagiarism Detection Software”, pp 556-561.

[3]. Segers V.M. andConnan J., “An Online System for Plagiarism Detection” pp. 1-4.

[4]. ChewE. andBlackey H., “e-Plagiarism Detection at Glamorgan” pp. 16-20.

[5]. Mozgovoy M.,Fredriksson K., White D., Joy M. and Sutinen E., Department of Computer Science, University of Joensuu, “Fast Plagiarism Detection System”. pp. 11-14.

[6]. Zou D., Long W.J., Ling Z., “Two-Phase Plagiarism Detection Method”, IEEE 2011.pp 1-4.

[7]. Anzelmi D., Carlone D., Rizzello F., Thomsen R. and Hussain D.M.A, “Plagiarism Detection Based on SCAM algorithm”, Proceeding of International MultiConference of Engineers and Computer Scientists 2011 vol. 1, IMECS, March 16-18, 2011. pp. 272-277

[8]. Ali A.M.E.T, Abdulla H.M.D and Snasel V., “Survey of Plagiarism Detection Methods”, Fifth Asian Modelling Symposium, 2011.pp. 39-42

[9]. Park S.Y., Kim S.Y., Kim S.H., Cho H.G., “A Global Dictionary Based Approach to Fast Similar Text Search in Document Repository”, 11th IEEE Conference on Computer and Information Technology, 2011. pp. 526-532.

[10]. Ali A.M.E.T, Abdulla H.M.D and Snasel V. and Vondrak I., “Using Kohen Maps and Singular Value Decomposition for Plagiarism Detection”, Third International Conference on Computational Intelligence, Communication System and Networks, 2011.pp. 60-64

[11]. en.wikipedia.org/wiki/Plagiarism_detection