IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 7, Issue 8, August 2018

A Survey: Document Spamming Detection and Plagiarism Content Techniques

Nidhi Ruthia[1], Abhigyan Tiwary[2]

M. Tech., Research Scholar, Department of CSE, SIRTS, Bhopal, M. P[1]

Assistant Professor, Department of CSE, SIRTS, Bhopal, M. P[2]

Abstract: Plagiarism is defined as representing someone else words, thoughts, knowledge, methods, programs etc in our own name. Plagiarism has a wider meaning by paraphrasing someone else text by replacing some data or method in our way is also a plagiarism. It also violates the rule if you don't mention someone author name when you are copying their data in your own way of representation. The detection techniques are applied by differentiating between variety of languages such as natural and programming language. From the existence of the previous approaches like plagiarism detection technique and SCAM analysis from the document stream. A further solution to find plagiarism in the given input data, and textual data is required. Here we are using QAP based Function minimization approach. This technique can be used to find document from the stream. Further QAP could enhance the required minimization function.

Keywords: Plagiarism, QAP, LCS, SCAM, Plagiarism Detection Techniques

I. INTRODUCTION

In today's era, world is moving so fast that no one has a time to create and publish its own format of data, instead follows the copy-paste approach from someone else data and somehow manipulate the same. What plagiarism actually means is to edit someone else knowledge into our format and pasting their own name. Plagiarism is not just mean to copy the data but also manipulating, converting, taking small parts, etc. It is difficult to manage when it comes to deal with scientific research work where same data is available throughout the internet. With the advent and highly usage of internet, peoples are simply copying the data from different websites and pasting by their own name, infect plagiarism allows to do copying with the fact that person should mention the real author name and their copyright. The plagiarism can be found in text documents as well as multimedia based documents, here we do concentrate on text documents more.

Plagiarism is becoming major task to solve world-wide and increasing as well. As new authors reuse the available data over the internet and add some of their own content without giving credit to the source author. There can be two types of plagiarism found in natural language and programming language. The detection methods and algorithms are different for both the languages, such as natural language focuses on textual features while in programming language, variables used, parameters, subprograms, and statements are followed. The plagiarism cannot be simply avoided by any direct algorithms or methods. To avoid plagiarism, we need to follow the two step criteria i.e., prevention and detection of plagiarism. As plagiarism prevention is difficult to obtain and needs long time to accumulate but sustain to long terms. In prevention, collaborative efforts are required to obtain and count the plagiarized data at every level. Prevention helps in minimizing the possibility of plagiarized content but quality remains the makeable issue. On the other hand, plagiarism detection can be carried out manually or by the help of computer software. Now days, software detection methods are likely followed as detection is easier, faster and results are more accurate than the manual detection.

Culwin and Lancaster's explained a four way process to detect plagiarism [1].

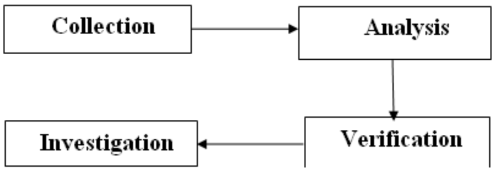

Fig.1 Four way process to detect plagiarism

- The collection stage defines the process of uniformly collecting the document and pre-processing the collected document into format suitable for further stage.
- Analysis is a stage where document is compared or analyzed with the existing one, documents obtained from web and list of those documents which are further required to be investigated.

DOI 10.17148/IJARCCE.2018.7818

- The verification stage is necessary as it ensures that the suspected document which undergone through investigation are accurately verified.
- The last stage of detection determines the probability of accuracy and fault free penalties.

II. PLAGIARISM DETECTION TECHNIQUES

Plagiarism is getting worse day by day as every content is available on web, thus it is a worldwide problem to solve and detect the plagiarism. There are two main plagiarism detection techniques which have their further categories based on their specific features. In order to detect plagiarism two methods are used such as Manual detection and Computer-Aided detection [2]. These two methods are further used upon the type of plagiarism whether using in textual based content or source code based content. The plagiarism detection techniques are briefly described in the fig.2

Manual Detection: This is the process of spotting manually the instances of plagiarized content within a document. This type of plagiarism detection is done manually by the expertise persons by comparing and verifying the given set of documents. The manual detection is generally done for the small documents verification as it is impractical on large document files. Class work, short notes, articles etc are example of this kind of plagiarism detection.

Computer Aided Detection: This type of detection method requires computer system to detect the plagiarized content with the help of suitable algorithms. Using manual techniques, the infeasible results are obtained while a computer-aided technique gives fast and accurate detection with the usage of specified algorithms. This technique is mainly implied in large document content so that quick and feasible outcomes can be found.

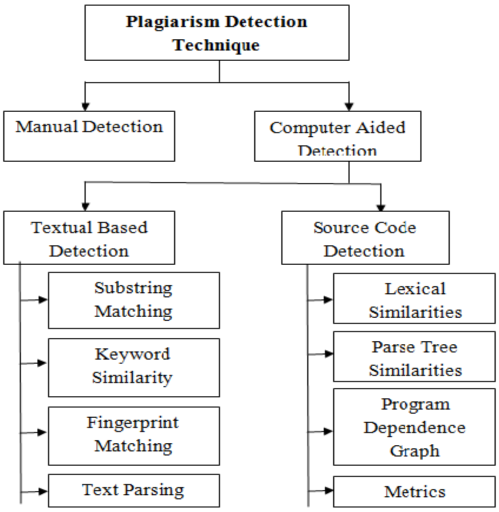

Fig.2 Classification of Plagiarism Detection Technique

1. Textual Based Plagiarism Detection Technique

Text based plagiarism is another word of copying other language from source, and has become a major issue of growing concept in scientific research and publishing. It can be intrinsic and extrinsic. In extrinsic plagiarism detection, the reference document is compared with all other documents while in intrinsic, the reference document is being compared with the external document. There can be either manual technique or computer based technique to detect plagiarism. As manual techniques are difficult to implement, there is a need to use computer based plagiarism techniques. Following are the various algorithms used in textual based plagiarism detection:

1.1 Substring Matching: In substring matching, the word substring automatically explains to broken down the string into substrings. In other words, the documents present to be compared are broken down into substrings and are stored in another list. The document is divided into substring using indicators such as ‘,’ , “,” , ‘?’ , etc then the list are compared through-out the available lists. If the substrings are found same, then the plagiarism count is increased [3].
The substring matching is mainly carried out through following matching techniques included their performances:

- Brute Force : Worst case time complexity is O(MN), Best case is O(M)
- Rabin Karp : Worst case time complexity is O(MN), Best case is O(N)
- Knuth Morris Pratt : Total time complexity is O(N+M)

Where N characters in text length, M characters in pattern length.

1.2 Keyword Similarity: In Keyword similarity method, keywords plays vital role. The document that needs to be compared is partitioned into phrases on the basis of keywords. Afterwards, on the basis of similarity between the

DOI 10.17148/IJARCCE.2018.7818

documents are calculated [3]. In other words, the keyword is first obtained and using this keyword we will find the sentence from both the document. Then compare both the sentences again, if found same then add to plagiarism set.

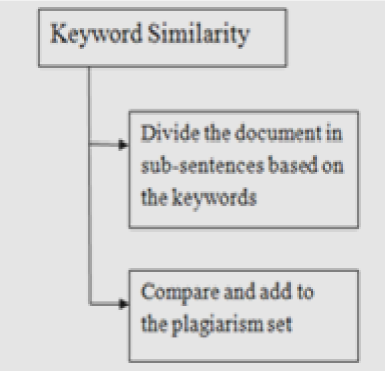
Keyword Similarity
Divide the document in sub-sentences based on the keywords
Compare and add to the plagiarism set

Fig.3 Keyword Similarity Structure

1.3 Finger-Print Match: Fingerprint matching has a wider meaning in detection of plagiarism. It considers the documents then scans the fingerprint of the documents. The name fingerprint is different in its meaning as it completely lies on the principle of k-grams solution where the document is partitioned into certain grams of k-length. Then the documents are compared on the basis of evaluated fingerprints hence the plagiarism is detected. Based on their correlation and similarity measures fingerprint matching techniques are divided into three category; character-based fingerprints, phrase-based fingerprints and statement-based fingerprints [4]. The fingerprint matching technique lies under the category of character-based plagiarism detection. In case of fingerprint computation, two main problems effects the hashing standards as they causes expensive computation cost and small size piece must be used in order to identify matching pattern.

1.4 Text Parsing: Text parsing is the method of analyzing the sequence, strings, symbol of any sentence used either in natural language or programming language with the help of predefined grammar [5]. By the end of analysis, parser obtains the data structure which is based on formal grammar and converts into tokens. Here, the data structure implies the meaning of structural format of sentence i.e. tree. The structure of sentence must be in the form of tree which is automatically generated and formed. For example, sentence 'John hits the ball' will be parsed as -

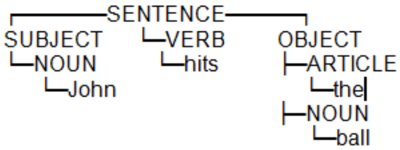

Fig.4 Tree structure for text parsing

After the formation of tree, the matching procedure gets started. Initially the flowchart based formulation is done for each file, and then algorithm performs a rough abstract comparison while keeping in note that only parse tree elements are being considered. This is followed for recursive way too at each level of tree nodes. The comparison is being noted, if indicates enough similarity a special micro comparison is done. Further, each tree nodes separate into sub tree that has to be compared with another relevant sub tree from another file [6].

2. Source Code Based Plagiarism Detection Technique

The plagiarism detection is quiet hard and time taking to find out the similarity or plagiarism in the source code. But detection of the source code plagiarism is a wider need and important for scientific research and academics. Since the students copies data from the web through source codes and paste it as own as shown in [7, 8]. The programming codes are as well copied from various internet sites, without keeping in knowledge of accurate format and genuine language. From [9,10] we came to know the plagiarism based on source code can be achieved using following algorithms like lexical similarities, parse tree similarities, program dependence graph, metrics, etc.

2.1. Lexical Similarities: To trace the apprehensive documents and to detect the plagiarism, lexical similarities involve the usage of lexical features of the text or documents. This lexical feature operates on the word level or character of the document [11]. The source code is converted into lexical tokens from where compiler will extract the meaningful content from the source. This type of approach mainly focuses on the similarity and comparison measures, there-by differ from one technique to another. The comparison unit differs in different techniques include word, sentences, passages, sliding window or n-gram. The processing technique upon which lexical similarities relies are tokenization, punctuation removal, lowercasing, grammatical stemming, etc.

2.2. Parse Tree Similarities: Using parse tree for plagiarism detection makes the detection easy and quick. The parse tree is being build from the lexical analysis therefore, illustrate the structure for programs. Both the structure for fragment of codes and the lexical streams lies in the parse tree. During compilation process, the compiler makes parse tree for each program and the algorithm for plagiarism detection do the same with parsing each program. After parsing, it finds common sub tree from each pair of parse tree. The parse tree is extracted with ANTLR which is a language tool that provides a translator for grammatical expression and framework for constructing compiler.

Let subtree1,subtree2,... be all of the subtree$_s$ in a parse tree T. Then, T can be represented as a vector, taken from [12, 13].

$$V_T = \langle \#\text{subtree}_1\ (T), \#\text{subtree}_2\ (T),..., \#\text{subtree}_t\ (T)\rangle$$

where #subtree$_i(T)$ is the frequency of subtree$_i$ in the parse tree T.

The kernel function between two parse trees T_1 and T_2 is defined as

$$K_{\text{tree}}(T_1, T_2)=V_{T1}\,{}^{t}V_{T2}$$

and is determined as $K_{\text{tree}}\ (T_1, T_2)=V_{T1}\,{}^{t}V_{T2}$

$$\begin{aligned}
&= \textstyle\sum_i \#\text{subtree}_i\ (T_1) \cdot \#\text{subtree}_i\ (T_2) \\
&= \textstyle\sum_i (\sum_{n1 \in NT1} I\text{subtree}_i\ (n_1)) \cdot \\
&\quad (\textstyle\sum_{n2 \in NT2} I\text{subtree}_i\ (n_2)) \\
&= \textstyle\sum_{n1 \in NT1} \sum_{n2 \in NT2} C\ (n_1, n_2)
\end{aligned}$$

where NT_1 and NT_2 are all the nodes in trees T_1 and T_2. The indicator function Isubtree$_i$ (n) is 1 if subtree$_i$ is rooted at node n and 0 otherwise. $C(n_1, n_2)$ is a function which is defined as $C\ (n_1, n_2) = \sum i\ I\text{subtree}_i\ (n_1) \cdot I\text{subtree}_i\ (n_2)$.

2.3. Program Dependence Graph: Through program dependence graph (PDG) source code is graphically represented in the program and the units like variables, assignment units, function calls are represented on the vertices of the graph. The edges are the connection between dependencies of the program. The PDG have mainly two components data dependency and control dependency edges [14]. Data dependency edge implies that the explicit representation between the existing relationship in the source program. Control dependency edge from vertices represents if the predicate is being evaluated on the present attribute of edges, then the second component of vertices will be executed.

2.4. Metrics: Software Metrics are used to analyze similarity measures in plagiarism by using functionalities like number of branches, loop, statement, parameters, function calls, user defined variables, local variables, global variables. The source code must be parsed to use through metrics and further fragments are computationally easy to calculate, comparison can be quick as well.

III. RELATED WORK

In December 2016 [15], Plagiarism detection method is being proposed by the author, which lies on the principle of local maximal value of the longest common subsequence (LCS) by its length and weight. They introduce three methods based on the following three document similarities: for two documents, • the length of LCS divided by the length of the shorter document, • the local maximal value of the length of LCS, and • the local maximal value of the weighted length of LCS. Therefore, the result shows that the proposed method was superior to the other two methods while taking some factors such as document similarity, plagiarism detection and accuracy, and Datasets.

In November 2016 [16], Shivani and Vishal Goyal suggested the idea of plagiarism detection system in English language. Further, they discussed textual based plagiarism detection on an exact string matching technique through the database and web. To implement their work, they performed three steps such as Pre-Processing where splitting the input string into individual sentence is done to obtain filtering of unnecessary words. The second process is sentence searching throughout the database and web, if plagiarized sentence is present in the DB then sentence is directly retrieved, otherwise cosine similarity approach is used for throughout searching on web. Finally, Similarity analysis is performed for detail description about all plagiarized sentences with the URL.

In this paper [17], author analyzes the evaluation of plagiarism detection, and uses previously researched concepts to organize present approaches on detection. The first situation which was taken into measure is extrinsic plagiarism detection where source document is considered. When a document is identified through its citation sequence but not by own text, then it follows citation based plagiarism detection approach. While in semantic role labeling method similarity between sentences are calculated. The cross lingual semantic approach uses a semantically annotated graph model for cross lingual plagiarism detection. Lexical and syntactic features are combined in regular cross lingual approach. Finally, they revise the current research situations and derive research methods to be followed further.

In this paper [18], author proposed a comparison between source and suspected text documents by using some text documents as a case study. They made the data ready by preprocessing, tokenization and Morphological analysis before

DOI 10.17148/IJARCCE.2018.7818

documents comparison. Further, they described novel trie based method to detect and save the source and suspected documents in solving the detection problems. The reasons to perform trie tree structure are the fast insertion and retrieval of long sentences in plagiarism detection problem. To evaluate the algorithm, they used macro-averaged precision and recall, granularity measurements, and the plagdet score which were proposed by the PersianPlagDet competition.

Daniele Anzelmi, D. M. Akbar Hussain, et al. [19] suggested detection process which is based on comparison of documents through natural language. The SCAM (Standard Copy Analysis Mechanism) is implemented to determine the similarity score of each pair of document, since SCAM is an approximate measure to detect overlapping between test document and registered document by making comparison on set of words that are common between both documents.

Dragan Gasevic [20] has reviewed many existing solution for source code similarity detection, where they got the clue that structure oriented code based approach which uses tokenization and string matching algorithms to detect source code similarity. But he proposed himself his approach for similarity detection as previous methods were not extensible and includes template codes. The proposed method consists of five phases, such as, pre-processing, tokenization, exclusion, similarity measurement and final similarity calculation. The pre-processing and tokenization is included to reduce the noise and to convert source code into tokens, these are programming language dependent. In exclusion, all findings of excluded token sequence are removed by using RKR-GST algorithm. RKR-GST (Running-Karp-Rabin Greedy-String-Tiling) and Winnowing algorithm are used to detect other phases.

In 2012, Catur Supriyanto, et al. proposes a comparison between Rabin Karp and Semantic based Plagiarism Detection as the similarity computations of two documents are required and their accuracy is measured. In place of Semantic based document plagiarism, author employed Latent Semantic Analysis (LSA) approach via Singular Value Decomposition (SVD). They uses dice similarity for Rabin Karp method while cosines similarity for LSA based document. For performance analysis they used intrinsic detection method and found that Rabin Karp is simpler than LSA and has better performance too. This evaluation has done on a data corpus of 100 documents and they plan further to evaluate perfomances on large corpus[21].

Here in table 1 below is the complete comparison of available work for the plagiarism detection taken from different references such as [15],[16],[17],[18],[19],20],[21].

Table 1. Different mechanism performed by the previous authors.

S. No	Paper Title	Name of Author	Approach Performed	Description
1.	Plagiarism detection using document similarity based on distributed representation Year- 2016	Kensuke Babaa, Et al.	Method based on local maximal length of Longest Common Subsequences (LCS) with weight.	Superior than simple length of LCS and local maximal value of LCS without weight
2.	A novel approach for plagiarism detection in English text Year- 2016	Shivani, Vishal Goyal	Textual based plagiarism detection	String matching is follow throughout the database and web.
3.	Plagiarism detection state of the art systems and evaluation methods Year- 2016	Christina Kraus	Evaluation of plagiarism detection	For evaluation, different approaches are considered
4.	Plagiarism detection based on a novel trie based approach Year- 2016	Alireza Talebpour, et al.	Novel trie based approach	Comparison of source and suspicious document.
5.	Plagiarism detection based on SCAM algorithm Year- 2011	Daniele Anzelmi, et al.	SCAM approach	To obtain similarity scores from each pairs of document.
6.	A Source Code Similarity System for Plagiarism Detection Year- 2013	Dragan Gasevic	Similarity Detection approach	To achieve better extensibility and inclusion of template codes.
7.	A Comparison of Rabin Karp and Semantic-Based Plagiarism Detection Year -2012	Catur Supriyanto , et al.	Similarity computation between Rabin Karp and semantic based approach.	It is found that Rabin Karp is simpler and has better performance than semantic based detection.

DOI 10.17148/IJARCCE.2018.7818

IV. SYSTEM CHALLENGES

In the system document processing and plagiarism finding, there are different challenges while occur while processing the document stream. The understanding document processing challenges are:

1. Working towards the multiple document extension and finding the similarity content from them.
2. Working with the similarity matching among the content with same meaning and semantic understanding of vocabulary.
3. Working towards the permission access of multiple online data availability and then processing them in document.
4. Find mathematical computation solution in document matching.

Thus the challenges requirement is need to be furnishing while developing solution to existing content matching algorithm.

V. FURTHER ENHANCEMENT

A QAP based Function minimization approach can be proposed where we can modify the existing technique by new and more efficient technique of data finding and collection as well as trend finding. We can replace some previous concept which is necessary for content searching aspect that will help to reduce computational time as well as total execution time.

CONCLUSION

From this paper, author tried to first describes the meaning of plagiarism, how plagiarism can be avoid, detect and prevent. Further the stage of plagiarism detection is being explained and those various techniques which are available to detect plagiarism are briefly described. Later on, the previously done works were discussed to know better and deeply about the concept from different research and review papers. Lastly, this paper is ended with the problem challenges which were observed during the research and some enhancement is proposed which could be done to improve plagiarism detection.

REFERENCES

[1]. Fintan Culwin and Thomas Lancaster, Staffordshire University, “Plagiarism, prevention, deterrence and detection”, uploaded on 2014.
[2]. Shameem Yousuf, Muzamil Ahmad, and Sheikh Nasrullah, **“**A review of plagiarism detection based on Lexical and Semantic Approach**”,** International Conference on Emerging Trends in C2SPCA, © IEEE 2013.
[3]. Manav Bagai, Vibhanshu, Siddharth Gupta, and Rashid Ali, “Text based plagiarism detection”, International Journal for Technological Research in Engineering Volume 3, Issue 8, April-2016.
[4]. Taiseer Abdalla Elfadil Eisa and Salha Alzahrani, “Existing plagiarism detection techniques: A systematic mapping of the scholarly literature”, June 2015.
[5]. Seong-Bae Park, et al., Korea, “Program Plagiarism Detection Using Parse Tree Kernels”, © Springer-Verlag Berlin Heidelberg 2006.
[6]. Sangeetha jamal, Cochin university of science and technology, Cochin, “Plagiarism Detection Techniques”, 2010.
[7]. Michal Duracika, Emil Krsaka, and Patrik Hrkuta, “Current trends in source code analysis, plagiarism detection and issues of analysis big datasets” TRANSCOM, 2017.
[8]. Kshitiz Gupta, Ekta Sardana, “Source Code Plagiarism Detection using Multi Layered Approach for C Language Programs” , International Journal of Computer Applications (0975 – 8887) November 2014.
[9]. Tapan P. Gondaliya, Hiren D. Joshi (PhD), and Prof. Hardik Joshi, “Source Code Plagiarism Detection ‘SCPDet’: A Review”, International Journal of Computer Applications (0975 – 8887) Volume 105 – No. 17, November 2014.
[10]. Georgina Cosma and Mike Joy, University of Warwick, Coventry, “Towards a Definition of Source-Code Plagiarism”, IEEE Transactions on Education, May 2008.
[11]. Shameem Yousuf, Muzamil Ahmad, and Sheikh Nasrullah, “A review of plagiarism detection based on Lexical and Semantic Approach”, International Conference on Emerging Trends in C2SPCA, © IEEE 2013.
[12]. Jeong-Woo Son, Seong-Bae Park, and Se-Young Park, Kyungpook National University, Daegu, Korea, “Program Plagiarism Detection Using Parse Tree Kernels”, © Springer-Verlag Berlin Heidelberg 2006.
[13]. Hyun-Je Song, et al. “Computation of Program Source Code Similarity by Composition of Parse Tree and Call Graph” , Hindawi Publishing Corporation, December 2014.
[14]. Jens Krinke, “Identifying Similar Code with Program Dependence Graphs”, University at Passau, Germany, IEEE.
[15]. Kensuke Babaa, Tetsuya Nakatohb, and Toshiro Minamic, “ Plagiarism detection using document similarity based on distributed representation” IAIT2016, 19-22 December 2016, Macau, China.
[16]. Shivani and Vishal Goyal Phd, “A Novel Approach for Plagiarism Detection in English Text” International Journal of Computer Application, November 2016.
[17]. Christina Kraus, “Plagiarism Detection - State-of-the-art systems (2016) and evaluation methods”, arXiv:1603.03014v1 [cs.IR] 8th March 2016.
[18]. Alireza Talebpour, Mohammad Shirzadi Laskoukelayeh, and Zahra Aminolroaya, “Plagiarism Detection Based on a Novel Trie-based Approach”, Fire 2016.
[19]. Daniele Anzelmi, D. M. Akbar Hussain, et al. “Plagiarism Detection Based on SCAM Algorithm”, International MultiConference of Engineers and Computer Scientists 2011, Hong-Kong.
[20]. Dragan Gasevic, “A Source Code Similarity System for Plagiarism Detection” , The Computer Journal, 2013.
[21]. Catur Supriyanto, et al. “A Comparison of Rabin Karp and Semantic-Based Plagiarism Detection”, 3rd International Conferences on Soft Computing, Intelligent System and Information Technology 2012.

DOI 10.17148/IJARCCE.2018.7818