Materials for
Ph.D. Scholars 19-20

RPE05 : PUBLICATION MISCONDUCT

B: SOFTWARE TOOLS

- Use of Plagiarism softwares like Turnitin, URKUND and other software Tools

— Please see the theory below and note down features of above tools from this website

— R Jaishankar
01.06.20

An Improved Continuous N-gram Technique for Textual Plagiarism Detection in the Research Documents

R.C. Tripathi NGB(DU) Prayagraj
Email: dean.engi@ng-bu.edu.in

Abstract: In the current information age the literature related to every subject matter is now available to large extent freely for access by everyone over the internet subject to attached copyright tags. In the academic and scientific research community, however such scientific and research publications must be plagiarism free. Any document submitted for the publication from an academic and research organization in general uses copy-free as well as the copyrighted documents. In the present paper, we propose a continuous trigram method and its application for the retrieval of the similar text from the local corpus as well as from the internet. This technique also catches the actually plagiarized chunks borrowed in the documents and not only words counts which are the state of the art of similar tools today. It is found to be efficient than all such tools reported as such.

Keywords: Textual Plagiarism, Information Retrieval, N-gram Technique, Textual Matching, etc.

1. INTRODUCTION

There are several issues in the recent years related to academic plagiarism. The use of copy-free contents available on the internet can be easily accessed by everyone and can be used for any academic or publishing purposes. However, most of the internet contents have a copyright notice on them. There is a need of some automated system by which the plagiarism can be found in the new textual matter received for publication. Various efforts have been made to solve this issue. Any publication whether it is academic or by a research organization, if it contains significant amount of other's research work as copied without giving due credits to such other author's work or any copyrighted material used, then it is called "plagiarism". The substantial amount of text, tables, software source code/pseudo code or other textual contents if copied this way, need to be detected before accepting a document for publication. The approach to develop some application software tool to automate such a system providing accurate results is the need of the hour. Such an automated plagiarism detection system should comprise several components including those to detect self plagiarism and contents of the references cited in the article in question. In the

present work, we propose a continued trigram methodology with its applications for offering such an automated plagiarism detection system in regard to the textual contents.

Detecting plagiarism and how to avoid academic plagiarism has been an area of interest since the long past. Denning (1995) suggested establishing libraries of academic works to avoid the plagiarism in the academic submissions for publication of the Journals or Conferences proceedings or the books. This seems to be a generally sensible idea, similar to the way in which fault detection services are of importance in areas like raw materials used by the factories. Samuelson suggested that 30% shared similarity is acceptable for self plagiarism (Samuelson 1994). In the current situation however, this is not accepted as such particularly if the past material some copyright transfer agreement has already been submitted by the authors to some other publishers. The similarity of the text should still be avoided for the word by word or exact similarity even when if it is same journal wherein the used material was even published in the same journal. It was also not clarified whether 30% means 30% of word count or 30% of page count. The pages would any way include diagrams, pictures, flow charts, tables *etc.* which are not counted as such in the textual plagiarism. Hence a more detailed examination of what constitutes self plagiarism is necessary to determine what appropriate prevention methods should deal with them.

Academia everywhere is making efforts to educate its work force the issues of the concerned plagiarism and how to avoid it. As a basic measuring criteria in the real practice to find whether two documents are similar or not, two basic approaches are now well established. These are i) the local (in the local repository) or direct and ii) global (in large set of data available on internet globally) or indirect approach (Ahlgren *et al.*, 2003; van Eck and Waltman 2009). In the local approach, the similarity between two documents is found by direct matching with every second document in the local repository. In the global approach, the similarity between two objects is obtained by measuring the similarity between their profile feature vectors that often contain the number of co- occurrences (eventually normalized) of an object with each other considered as objects (Cristian Colliander, 2011).

Lancaster and Culwin in year 2005 tried to classify metrics used for plagiarism detection. In their proposed methodology, they defined the metrics in two simple ways. First classification is based on the number of documents involved in metrics calculations process and second is based on

computational complexity of the design and methodology for similarity check up. In another approach of n-gram technique proposed by [S. Brin et al, 1995] emphasis is on the common string in two texts where these text are characterized with sequence of N consecutive characters. In these applications of statistical measures, each document can be defined with so called fingerprints [Stein. B et al,2006; Schleimer S et al 2003], where n-grams are hashed and then some are selected to form fingerprints. The probability based measures have been defined as information theoretical measure was advocated by [Aslam J.A et al, 2003], and language model measure was advocated by [Zhai C et al, 2001]. In the present paper, a variant of N-gram, (trigram) methodology has been proposed on a continued basis for output generation. The system is not only based on word count, it highlights the plagiarized portions in the output/results giving complete plagiarism report with all the copied text along with their source in a side by side two columnar display.

2. PRIOR ART OF THE WORK

In the previous decade, many techniques and methodologies have been reported for having been used for realizing an automated system for detection of plagiarism. Some of them were restricted only for the local repository as search space and some others were for web applications. In the natural course of plagiarism detection, some of them included detecting similarity across multiple texts as well as within one textual content. The plagiarism detection across multiple texts included searching for matching common substrings of length n, where n is chosen based on certain considerations [Brin. S et al, 1995; Shivkumar N et al, 1996; Lyon. C et al, 2001; Broder A Z, 1998]. If n is made fixed then the substrings are said to be n-grams. In the n-gram technique the substring can be restricted on any number of strings depending upon the algorithm used. The value of n may be different when retrieving subsequences from different parts of the document. The value of n however cannot be big since not all content is usually copied verbatim from a single source document.

Many methods have been reported for plagiarism detection. These include, for example, similarity between texts based on the longest common subsequence, approximate string matching, the overlap of longest common substrings (eg: COPS [Brin. S et al, 1995], Koala [Heintz N, 1996], YAP3 [Wise M, 1996] SCAM [Shivkumar N et al, 1996], JPLAG [Prechelt L et al, 2000], the proportion of shared content words, particularly those occurring only once,

CopyCatch [Woolls D et al, 1998], the overlap of consecutive word sequences or word n-grams (e.g. Ferret [Lyon. C et al, 2001], and compressed versions of the texts [Medori J et al, 2002] are some noteworthy developments noted in the history. Methods of detection originating from file comparison, information retrieval, authorship attribution, compression and copy detection have all been applied to the problem of plagiarism detection [Cloud P.D, 2003]. Methods have also been developed to visualize the similarity between texts including Visualisation and Analysis of Similarity Tool (VAST) [Culwin and Lancaster 2004]. Dotplot [Church K W et al, 1993], Bandit8 and Duploc [Ducasse S et al, 1999] are also worth mentioning. In another methodology of the text matching of the submitted document, statistical method used for detecting plagiarism makes use of "Latent Semantic Analysis (LSA)". In using the LSA, the word similarity and the extraction of the word sense or meaning and then their comparison for the similarity check in the body of the text is the key idea. This kind of system which detects semantic similarities to grade some work can also be used effectively for paraphrased plagiarism detection. Thus, the semantics of the words are recognized in detecting the plagiarism. The cosine similarity measure can then be used to find semantic relevance among passages at a much reduced computational cost.

Adams and Meltzer proposed [Adams E et al, 1993] trigrams and inverted files for exact matches with query terms. They reported [Canvar W B, 1994] 100% recall with high precision for their experiments and recommended trigram based search as an acceptable alternative to word-based search and a superior method for retrieval of common word cluster fragments. N-grams are used as an effective metric for TREC-2's retrieval and routing tasks providing promising results. Since N-grams are [Canvar W B, 1995; Cohen j D 1995] language-independent, the strategies used for retrieval can be used for document collections in languages other than English. N-grams are used [Lee J H et al, 1996] along with word-based systems for effectively retrieving compound nouns in Korean documents. N-grams can be used [Canvar W B et al, 1994; Damashek M, 1995] to distinguish between documents of different languages in multi-lingual collections and to gauge topical similarity between documents in the same language. Retrieval based on N-grams is found [Canvar W B, 1995; Huffman S, 1996; Robertson et al, 1992] to be robust to spelling errors or differences and garbling of text. In another experiments of fingerprinting system [A Chowdhury et al, 2002] proposed an intelligent system called I-Match, to filter out terms based on inverse document frequency. In the pre-processing of the developed

system, the most frequent and rarest terms are both removed. They used ranking method and weights to recognize the importance of document terms according to the frequency. The system focused on detecting near exact copies.

The earliest known systems for plagiarism detection were based on feature vectors. Ottenstein 1976 came up with such a system in 1976. Later some more systems were reported but their performance was not so remarkable. The System Developed by [Saul Schleimer et al 2003] named MOSS (Measure of Software Similarity) used for detecting plagiarism in computer software codes uses Rabin-Karp Algorithm with Windowing. However it wasn't of much use for detecting content plagiarism as it failed to detect the semantics associated with the document.

The present system has the distinction that it can test plagiarism from existing documents on the Internet as well as from the local repository of documents. The majority of softwares currently available addresses only one of above catchment areas and performs only the word or fingerprints set of data categories. Also the majority of software's currently available in the market are detecting program source texts in number of word format. The present proposed system is based on sentences and paragraphs chunks and its performance is much superior to the tools available for the purpose e.g VeriGuide [*www.**veriguide**.org*], DocCop [*www.**doccop**.com*], Plagiarism Detect [***plagiarism-detect**.com*] etc.

3. IMPLEMENTATION OF METHODOLOGY

Plagiarism is the most popular problem in the academic and publications area and it becomes a major challenge to train new authors how to keep their research articles free from plagiarism. A number of applications have been developed and day by day some of them are being updated. In the literature we have mentioned some of the developed applications and products which have some significant presence amongst the users for the said problem.

In the methodology proposed by us, we have adopted altogether a different approach of detecting the textual plagiarism, it does not use the set of rules of word matching or fingerprinting. Finding of maximum number of common textual chunks in the query research document and repository documents is the main target of our proposed methodology. We make use of a continuous trigram technique to match the textual data of query document with each document in repository and find the longest sequence and then calculate the similarity measure for plagiarism. This

system works well both on local repository as well as on internet. The system detects the longest sequence of the sentences of the two documents as long as it is continued right from the extent of word to line and beyond of the sentences. Then it counts the similarity of the text and displays the results in the output. The whole working architecture of the proposed system is given below in the figure1.

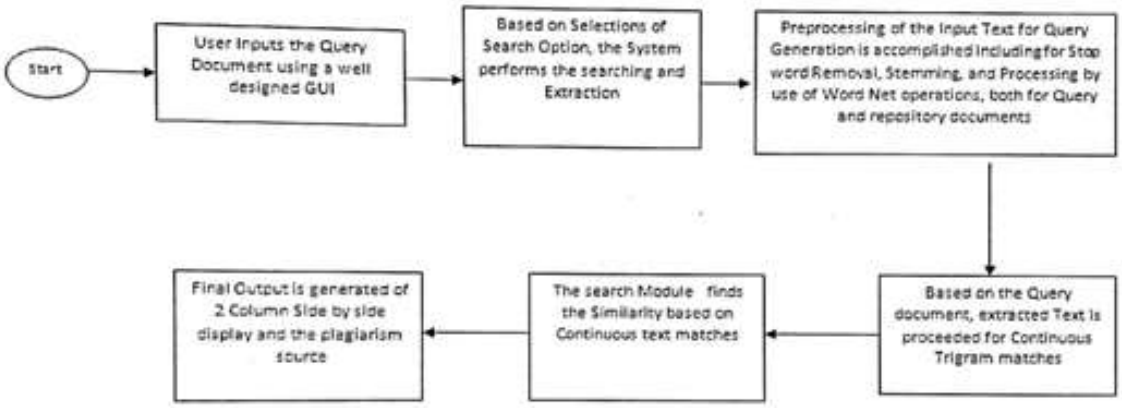

Figure1. The process diagram of the proposed methodology

3.1.Pre-Processing of the Input Document

The query input file can be uploaded in .pdf format for the plagiarism checkup. The system has been designed in such a way that a .pdf file will be converted in .txt file for further processing by the system itself. System asks the user to choose the option for the plagiarism checkup, whether the input query files will check on the local database or in the internet. The user has to specify the search option along with the query documents and only then the plagiarism checking process proceeds further.

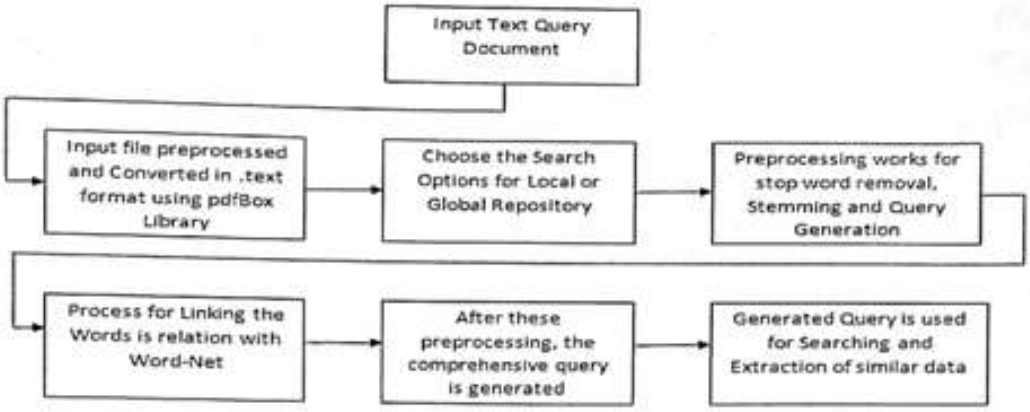

Figure 2 shows the flow chart of the preprocessing of the input data

In the proposed methodology of the plagiarism detection, there are two main modules. i) First is local data based plagiarism checkup and ii) Second is Online Internet based plagiarism checkup. In both of the proposed module, there is option of specific search, that means in the local database we have given a option to the user to search the plagiarism with specific files only and generate the plagiarism results. In the same line in the internet based search we have given option to the user to opt some free websites like Wikipedia or Encyclopedia web sites for free from searching and plagiarism criteria. In the initial steps for the plagiarism checkup, user can follow these criteria and options for the better and optimum result generation for a particular input query file. The system will perform the initial steps of stemming and finding the root word of the each and every word in the query input document with the help of wordnet after due removal of the 'stop words'.

3.2. Text Extractions for the Similarity Checkup

For the extraction of the similar text on the internet or from local repository documents, we use query file to provide the input document. The input file is first cleaned of all stop words and then the 'stemming' process enables to store all words in their root forms. Now, we generate query of trigram like fragments of words 1-3, 4-6, and so on and then search these entire generated query for the similar content on the internet with the help of search engines. In this process, we get the links of the similar contents and we download the respective contents from the results so obtained in the searching process. We save the downloaded documents in the local database in the .txt form for further processing. All the URL's so obtained are used to download relevant

contents to form a repository. In the searching process, one needs to neglect some copy-free web sites. The system creates a look-up list of such website addresses and when so ever, the link extraction process starts and encounters such addresses, the system skips the links and moves to the next URL to download the contents from such links.

For finding the suspected plagiarized portions from the downloaded documents, the sentences into trigram words sequences are fragmented. Let the suspicious document *s be* split into sentences (*si*). Now *si* is split into word *n*-grams. The set of *n*-grams represent the sentence. Thus a document *d* is not split into sentences, but simply into word *n*-grams; and each sentence $si \in s$ is searched singleton over the downloaded documents. In order to determine if *si* is a plagiarised portion from $d \in D$, we compare the corresponding sets of *n*-grams.

$$C(si \mid d) = \frac{|N(si) \cap N(d)|}{|N(si)|} \quad \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

where $N(\cdot)$ is the set of *n*-grams in $(\cdot)$. If the maximum $C(si \mid d)$, after considering every $d \in D$, is greater than a given threshold, *si* becomes a portion plagiarized from *d*.

3.3. Textual Similarity for the Input Document

In the proposed methodology, the main task is to search the similar text with all the downloaded documents from the internet or from the local repository of such documents and then generate the plagiarism results. For this purpose, the main query document is divided into several paragraphs with some words counts. The downloaded documents are divided into several paragraphs also with some words counts. The division of the paragraphs is also applicable for the local repository documents if the searching is specified on the local repository search. Now the searching module is invoked to compare the query document with the downloaded documents for the textual similarity.

In fact, each paragraph of the query document containing some set of words is compared with the set of two successive paragraphs in the downloaded or local database documents. We compare all the possible sequence of paragraphs from all the downloaded documents and we store such paragraphs which are textually more similar. We compare the corresponding paragraphs using the "Longest Common Subsequence Algorithm" (LCSA) to get the common

data of the two paragraphs. This is done to handle the para-phrasing concept which involves searching and comparing the given chunk by considering all the possible positions of words. The Pseudo code for the algorithm of the above process is as follows.

```
Get input document
Func Trigram (){
String line, line1, line 2, line 3, result
Perform Pre-processing of the input text by removing all stop words and stemming
Divide the input text in several paragraphs
        for (String ngram : ngrams(3, line)){
            String[] words = ngram.split(" ");
                for (String ngram1 : ngrams(3, line1)) {
                    String[] words1 = ngram1.split(" ");
    if(words.length==3 && words1.length==3)
if(words[0].equals(words1[0]) &&words[1].equals(words1[1]) && words[2].equals(words1[2]) )
{
  Result+=words[0]+" "+words[1]+" "+words[2];//append the match window to the result}
        Else {
            {//Match the antonyms synonyms using wordnet
```

The time complexity of the algorithm is 0(n*m) where n=database size and m==size of plagiarized data

4. RESULTS

The system has been designed with the user friendly environment. Figure 3 given below shows the graphical user interface designed for the query input by the user. On obtaining, the input file for the plagiarism search, it provides different options to search the input file for the plagiarism detection such as for the local repository search, web search and specific search as the catchment area to the user for the plagiarism detection. Figure 4 shows a typical output of the system. The

figure 4 given below shows the plagiarism detection results with the system designed output criteria i.e. side by side columnar display of query document sections along with the contents wherefrom it is copied as plagiarised portions. The efficiency and results output of the system have been tested on several input documents mainly research papers. The results of the system are most satisfying and effective. For this purpose, 50 research papers having different level of percentage plagiarism were evaluated. The percentage plagiarism found out varies form case to case in line with the initially modified documents for the testing purpose. Even almost 100% plagiarism cases were detected. The test results are summarized as table 1 given below. Figure 5 given below shows how % of plagiarism cases fall down for higher plagiarism infections in a typical newly established institute.

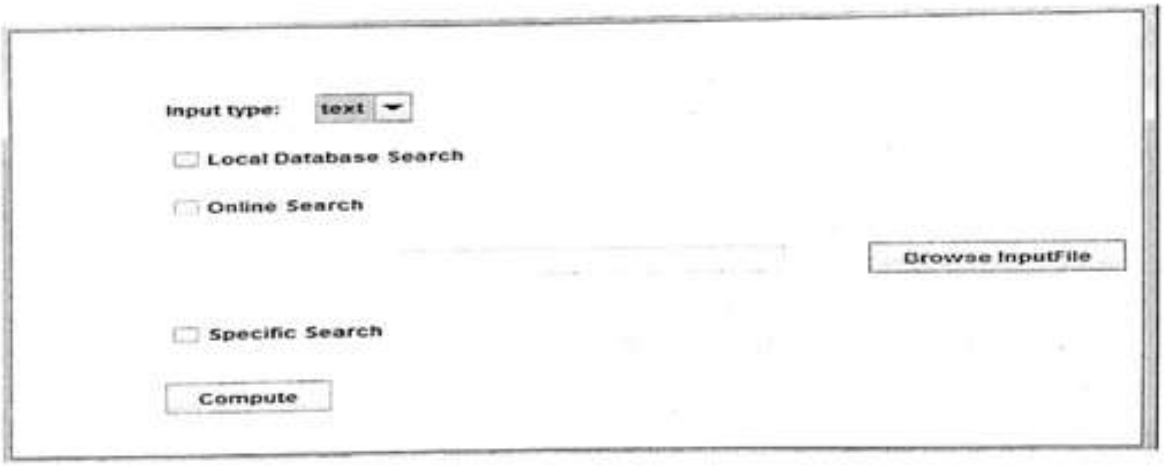

Figure 3 Shows the User Interface for inputting the test Data

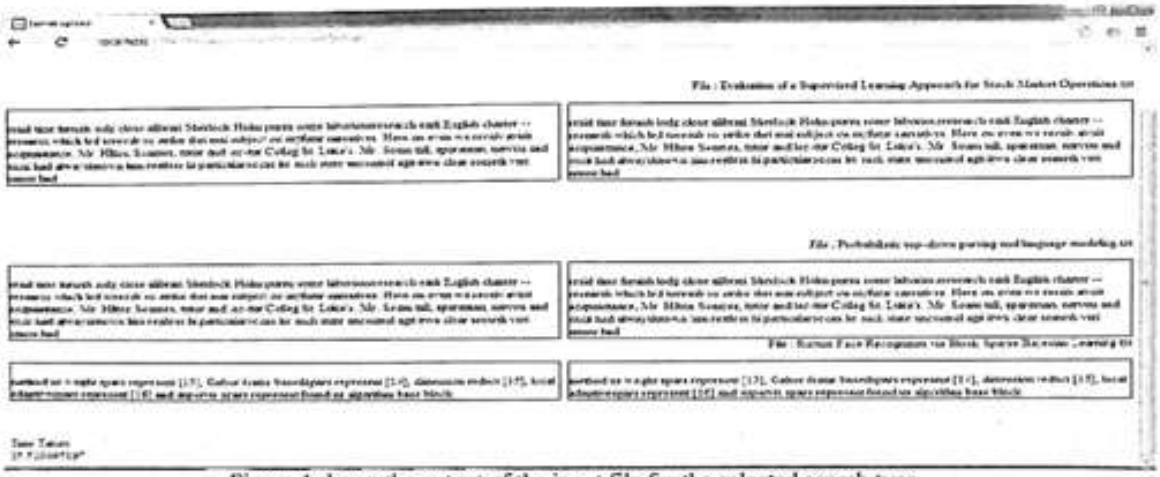

Figure 4 shows the output of the input file for the selected search type

S.No.	Percentage Plagiarism	Number of Cases	Overall % Share in total %
01	20%-30%	30	60%
02	30%-40%	10	20%
03	40%-50%	04	8%
04	50%- Above	04	8%
05	100%	02	4%
		Total- 50	

In the sample of 50 test cases, 30 research articles i.e. 60% of total were found to have plagiarism in the range 20%-30%. The graph plotted for the pattern of the test cases is shown in the figure 5 given below and shows that higher the level of plagiarism, smaller are the corresponding number of research articles i.e the number falls almost exponentially. Very few cases were found to have plagiarism in range of 40% to 50% type. Some of the cases were also found in which word for word is copied from the source and thus they manifested 100% plagiarism.

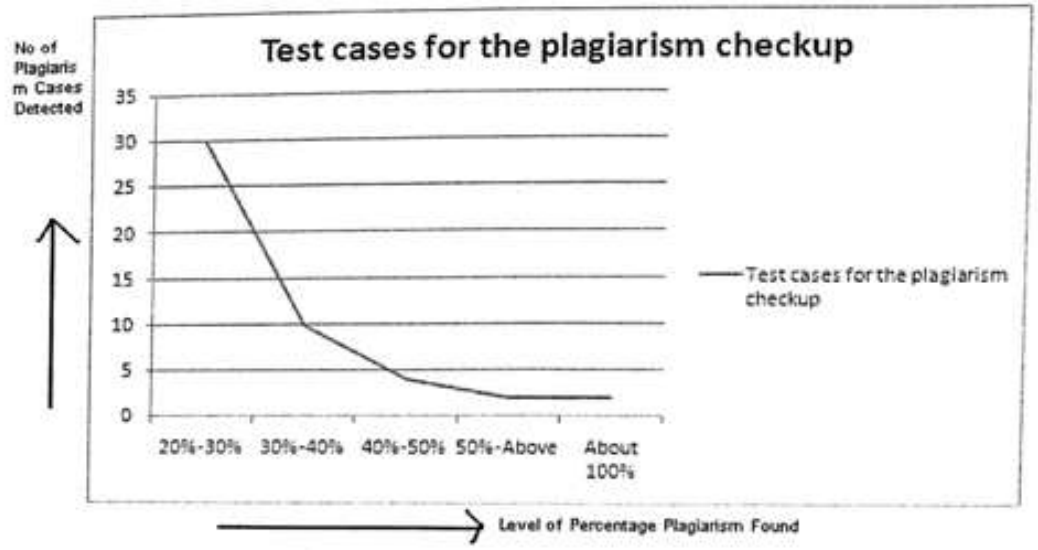

Figure 5 shows that the test cases prepared for the different level of plagiarism percentage in the query documents

After analyzing the outcomes of plagiarism test cases, three main patterns were noted. These patterns emerged since different types of documents contain different type of plagiarism. In most

of the research articles, the plagiarism type is paraphrasing i.e.; it contains less amount of exact copying material. A distinct pattern was seen in the larger documents such as Doctoral or Master Thesis. Here the infected documents contain exact copying or almost similar data copying. Performance of our tool was compared with some other tools for same test cases. This comparing table 2 is given below. In the given table 2, out of 30 cases shown in table 1, 10 plagiarism cases has been taken which has been tested with the developed software and found 20%-30% plagiarism which is already shown in the table 1. The above found 10 plagiarized document has been taken as input query file for the plagiarism detection from online available well known plagiarism detection softwares. They provide guest permission to check the plagiarism which has been used for the plagiarism checking process.

Table 2 Shows the results obtained from online available plagiarism detection software of 10 test cases

Some Online Plagiarism Checking Products	**(Percentage Plagiarism) 10%-20%**	**(Percentage Plagiarism) 20%-30%**	**(Percentage Plagiarism) 30%-40%**	**(Percentage Plagiarism) 40%-50% and Above**	**(Percentage Plagiarism) Free**
VeriGuide	04 40%	02 20%	02 20%	-	02 20%
DocCop (File Check)	02 20%	-	-	-	08 80%
Plagiarism Detect	03 30%	-	04 40%	-	03 30%
Plagiarism Tracker	02 20%	-	-	-	08 80%
Turnitin	01 10%	-	04 40%	05 50%	-
Total	12	02	10	05	21

The all 10 test cases have been tested in the online plagiarism checker product and find out some results. The Veriguide, Turnitin and DocCop plagiarism detection software detected some extent of plagiarism in all 10 cases but maximum number of cases found plagiarism in 10%-20% or plagiarism free. Few have detected having 30%-40% of percentage plagiarism. The DocCop software available online only file check process is free, so the results may be vary in this case but other than this available free for users.

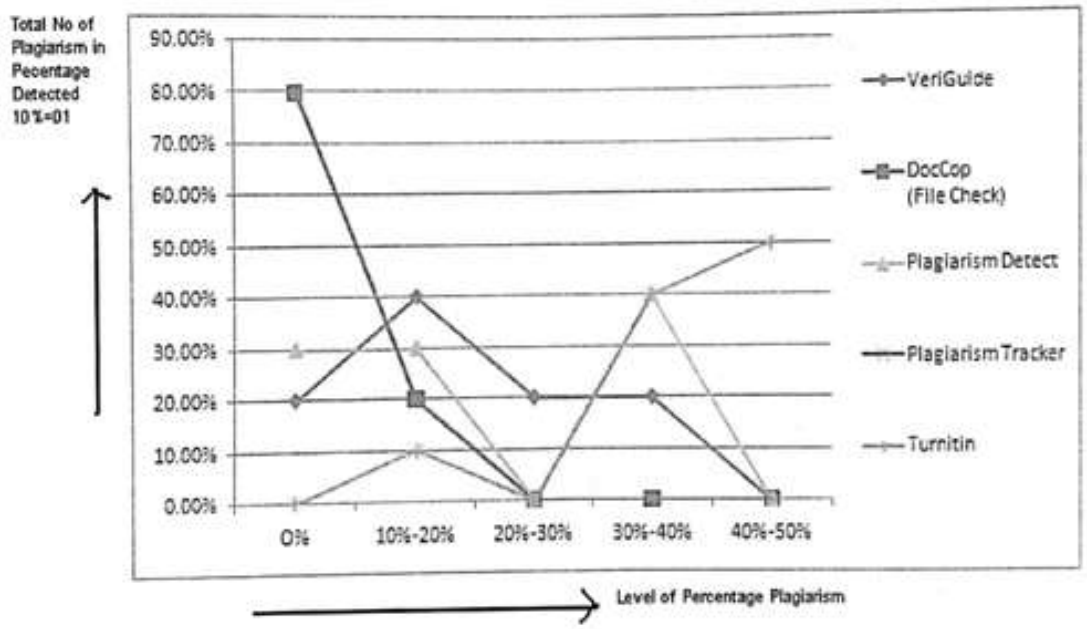

Figure 6 shows that the test cases prepared from online available 5 known plagiarism detection softwares

10 test cases have been taken from the above results given in the table 1 which is having 30%-40% plagiarism in each documents. Total test performs in four online software tools are 40. Out of 40, only 18 test cases found having plagiarism in some extent and 22 test cases found free from plagiarism. 14 test cases found having 20%-30% of plagiarism and 4 test cases found 40%-50% plagiarism which is checked by ***Plagiarism detect*** software tool. In the individual 10 test cases for each softwares ***Veriguide*** software tool detected 7 test cases having plagiarism 20%-30%. In figure 7 added the proposed method test cases which have level of plagiarism 20%-30% which is tested for others software in table 2.

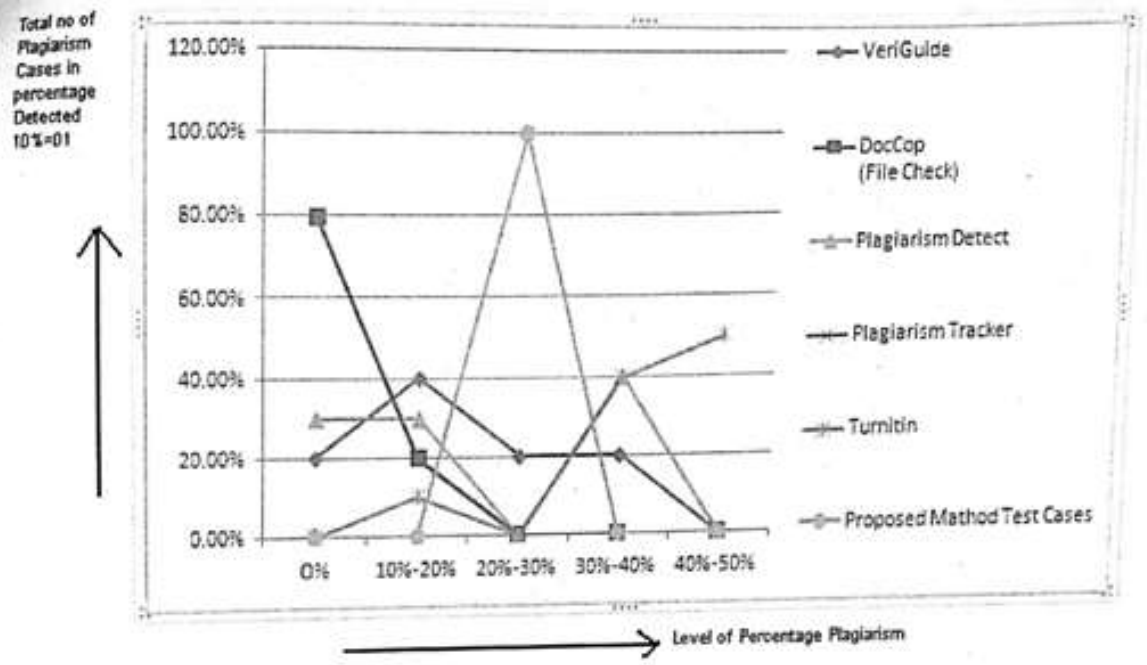

Figure 7 shows that the test cases prepared from online available 5 known plagiarism detection softwares with proposed method technique

5. DISCUSSION and CONCLUSION

In the current academic scenario, it is a challenge to protect the digital data and assure the originality in a new research paper intended to be published. Plagiarism in the textual data received for the academic or research publications are therefore a need of the hour. A number of applications have been developed in this regard so as to avoid plagiarism. Some of these major applications developed have been discussed in the section 4 of the current paper. There are different user groups of target documents needing the check-up of the plagiarism. Some of them are only for students' assignments and essays written in the pre-university classrooms. Whereas others are like book publishers, research conferences, journal paper etc. Based on the target group of users, some techniques were developed like trigram, variable words count, keyword overlapping, shingling, and visualization etc.

The recent research and developments in the area of finding textual similarity of the submitted document with those in a corpus is advancing day by day. Some of the professional setups in

these recent years are providing the services to the universities and research organizations for reporting the textual similarity of their submitted documents. The technologies and methodologies used in these developments for the similarity findings are mostly on the word counts. The word counts in the submitted documents with the similar corpus documents are used to provide the similarity index for the given text. Most of the cases are based on the corpus or local repository based resource documents. Web based repository has also been tried largely. However in both of these cases, the systems provide only a coarse value of plagiarism results and are not satisfactory.

In the current paper, the modified technique of trigram has been used which counts the longest sequence of the similarity matches and then catches the plagiarism portions in a chunk of data with paraphrasing applications. The results of our system have been tested and it is found to be superior to those as state of art tools like VeriGuide, DocCop, turnitin, Viper, Plagiarism, Plagiarism.net, Plagtracker, DustBall, DupliCheck, and Plagiarism Detect etc. During In the test cases evaluations, three major types of plagiarisms documents have been found. The results analyzed of exact copied plagiarized documents, almost plagiarized documents, and paraphrased documents. Each type of plagiarized documents has different set of conditions to retrieve such plagiarized documents. The more emphasis given to the time taken in the process of the input data for the plagiarism check up. In regard to the accuracy of the similarity checkup our tool not only checks similarity based on 3 words but it counts the longest words chunk and then decide to have plagiarism or not and then show in the results section.

The present system of the textual plagiarism detection is based on practical experiences and is a modified version of the main concept of the trigram technique for the selection of words to be used for the similarity check up in the documents. The main idea of 3 words has been modified to different level of words continued as per the need for realising improvements in the results. The system is found to be superior in its performance compared to popular competing tools used for the purpose of plagiarism detection in research documents.

REFERENCES

1. Lancaster, T., F. Culwin. Classifications of Plagiarism Detection Engines. ITALICS Vol. 4 (2), 2005.
2. Brin, S., J. Davis, H. Garcia Molina. Copy detection mechanisms for digital documents. Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, San Jose, California, pp. 398–409,1995.

3. Stein, B., S.M. Zu Eissen. Near Similarity Search and Plagiarism Analysis. Proceeding of 29th Annual Conference of the German Classification Society, pp. 430-437, 2006.
4. Schleimer, S., D.S. Wilkerson, A. Aiken. Winnowing: Local Algorithm for Document Fingerprinting. Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 76-85, 2003.
5. P. Samuelson, "Self Plagiarism or Fair Use, Communications of the ACM, August, vol. 37, no. 8, **(1994)**, pp. 21-25.
6. Peter. J. Denning, Plagiarism on Web, Editorial, Communication of the ACM December 1995/Vol. 38, No. 12, Page 29.
7. P. Ahlgren, B. Jarneving and R. Rousseau, "Requirements for a co-citation similarity measure, with special reference to Pearson's correlation coefficient", Journal of the American Society for Information Science and Technology, vol. 54, no. 6, pp. 550-560, **(2003)**.
8. N. J. van Eck and L. Waltman, "How to normalize co-occurrence data? An analysis of some wellknown similarity measures", Journal of the American Society for Information Science and Technology, vol. 60, no. 8, **(2009)**, pp. 1635-1651.
9. Cristian Colliander and Per Ahlgren, "Experimental comparison of first and secod-order similarities in a scientometric context", Scientometric Springer Publications, vol. 90, **(2011)**, pp. 675-685.
10. Aslam, J.A., M. Frost. An information-theoretic measure for document similarity. Proceedings of the 26th international ACM/SIGIR conference on research and development in information retrieval, pp. 449–450, 2003.
11. Zhai, C., J. Lafferty. A study of smoothing methods for language models applied to ad-hoc information retrieval. Proceedings of the 24th annual international ACM/SIGIR conference on research and development in information retrieval, New Orleans, Louisiana, United States, pp. 334–342, 2001.
12. Brin, S., Davis, J. And Garcia-Molina, H. (1995), Copy Detection Mechanisms for Digital Documents, Proc. of the ACM SIGMOD International Conference on Management of Data, 398-409.
13. Shivakumar, N. and Garcia-Molina, H. (1996), Building a Scalable and Accurate Copy Detection Mechanism, Proceedings of 1st ACM Conference on Digital Libraries DL'96.
14. Lyon, C., Malcolm, J. and Dickerson, B. (2001), Detecting Short Passages of Similar Text in Large Document Collections, In Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing, 118-125.
15. Culwin, F, Lancaster, T. 2004. Plagiarism prevention and detection. online. Available: http://cise.lsbu.ac.uk (Accessed 20th April 2005).
16. Broder, A. Z. (1998), On the resemblance and containment of documents, Compression and Complexity of Sequences, IEEE Computer Society.
17. Clough, P.D. (2003), Measuring Text Reuse, PhD thesis, University of Sheffield CopyCatch product website,http//www.copycatchgold.com/
18. Wise, M. (1996), YAP3 Improved Detection of Similarities in Computer Programs and Other Texts, Presented at SIGCSE'96, 130-134
19. Prechelt, L., Malpohl, G. and Philippsen, M. (2000), JPlag Finding plagiarisms among a set of programs, Faculty of Informatics, University of Karlsruhe, Technical Report 2000-1.
20. Woolls, D. and Coulthard, M. (1998), Tools for the Trade, Forensic Linguistics, Vol. 5(1), 33-57.
21. Heintze, N. 1996), Scalable Document Fingerprinting, In Proceedings of the Second USENIX Workshop on Electronic Commerce.
22. Medori, J., Atwell, E., Gent, P. And Souter, C. (2002), Customising a Copying-Identifier for Biomedical Science Student Reports Comparing Simple and Smart Analyses, M. O'Neill et al. (Eds), AICS2002, LNAI 2464, Springer-Verlag, 228-233.
23. Church, K.W. And Helfman, J.I. (1993), Dotplot A Program for Exploring Self-Similarity in Millions of Lines of Text and Code, Journal of Computational and Graphical Statistics, Vol. 2(2), 153-174.
24. Ducasse, S. and Rieger, M. and Demeyer, S. (1999), A Language Independent Approach for Detecting Duplicated Code, Proceedings ICSM'99 (International Conference on Software Maintenance), IEEE, 109-118.

25. Adams, E. S., Meltzer, A. C., *Trigrams as Index Elements in Full Text Retrieval Observations and Experimental Results*, ACM Computer Science Conference, February 1993
26. Cavnar, W. B., *N-gram-Based Text Filtering for TREC-2*, The Second Text Retrieval Conference (TREC-2), February 1994.
27. Cohen, J. D., *Highlights: Language- and Domain-Independent Automatic Indexing Terms for Abstracting*, Journal of the American Society for Information Science, 46(3), 1995.
28. Lee, J. H., Ahn, J. S., *Using n-grams for Korean Text Retrieval*, 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1996
29. Cavnar, W. B., Trenkle, J. M., *N-gram-Based Text Categorization*, Symposium on Document Analysis and Information Retrieval, April 1994.
30. Damashek, M., *Gauging Similarity with n-grams: Language-Independent Categorization of Text*, Science, Volume 267, February 1995.
31. Huffman, S., *Acquaintance: Language-Independent Document Categorization by N-grams*, The Fourth Text Retrieval Conference (TREC-4), October 1996.
32. Robertson, A. M., Willett, P., Searching for Historical Word-Forms in a Database of 17th-Century English Text using Spelling-Correction Methods, 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1992.
33. A. Chowdhury, O. Frieder, D. Grossman, and M. McCabe. Collection statistics for fast duplicate document detection. *ACM Transactions on Information Systems*, Vol.20, No2, April 2002, Pages 171-191
34. Ottenstein – "An algorithmic approach to the detection and prevention of plagiarism" SIGCSE Bulletin, vol.8, no.4, pp.30–41, 1976
35. Saul Schleimer, Daniel S. Wilkerson and Alex Aiken on the Winnowing: Local Algorithms for Document Fingerprinting (MOSS). *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*