Special Issue Published in International Journal of Trend in Research and Development (IJTRD), ISSN: 2394-9333, www.ijtrd.com

Plagiarism Checker

Vijaylakshmi Monisha and Naghma khan,
Department of MCA Department of Computer Science, Mount Carmel College, Autonomous Mount Carmel College, Autonomous, Bangalore, India

Abstract: Detection of plagiarism has become an important task for educational institutions, industries and for research organisations. This is because it has become very easy to cut and paste from the wide range of the free information available on the internet. Teachers need a plagiarism detection tool for the assessment of the assignment, as students tend to share assignments and collaborate with other students. The plagiarism detection tools are developed with a common goal to detect the copied material irrespective of the concepts and techniques that are used. This paper deals with the system that is developed by combining the concept of n-grams and substring matching. This system will be helpful in detecting the amount of plagiarism between two text documents.

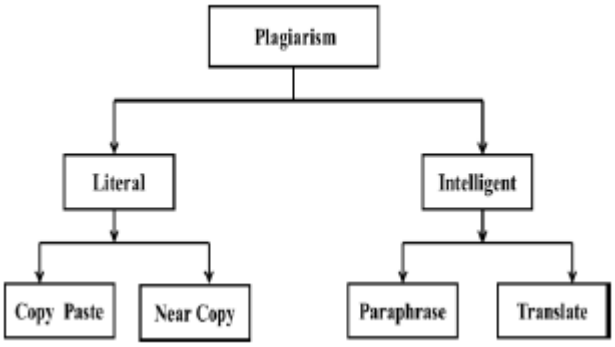
Keywords: Plagiarism detection, n-grams, substring matching

I. INTRODUCTION

Plagiarism is described as the act of taking other persons' writings or ideas and passing it off as your own. This can include information from web pages, books, articles or any other medium. By the tremendous growth of the internet, it is easy to find information. In general, plagiarism usually happens in textual documents. These could be essays, paper reports, students' assignments and sometimes code of programming languages [1].Detection of these becomes very tedious. This paper will talk about the types of plagiarism, the methods of detection and the concept that are used for the system.

II. TYPES OF PLAGIARISM AND METHODS OF DETECTION

Plagiarism comes in various types along with the various methods for detection [2].The figure 1 shows the types.



We have classified it into two main categories viz. *literal plagiarism* and *intelligent plagiarism.* a) Literal plagiarism can be understood as copying and pasting the contents without any modifications whereas b) intelligent plagiarism is when one transforms the copied material to make it look unique. Literal plagiarism further classified into *copy-paste plagiarism*- a type plagiarism in which individual just copies the contents from the source. *Near copy plagiarism* –this type is where the individual tries to remove some bits to make it look unique. Intelligent plagiarism is classified into *paraphrasing* – where the rephrasing of the sentences is done. Moreover, *translate plagiarism*, in here when rearrangement of the sentences is done then it may be said as Monolingual plagiarism. In addition, when the same work published in a different language it is multilingual plagiarism.

Figure 2. Shows the detection types

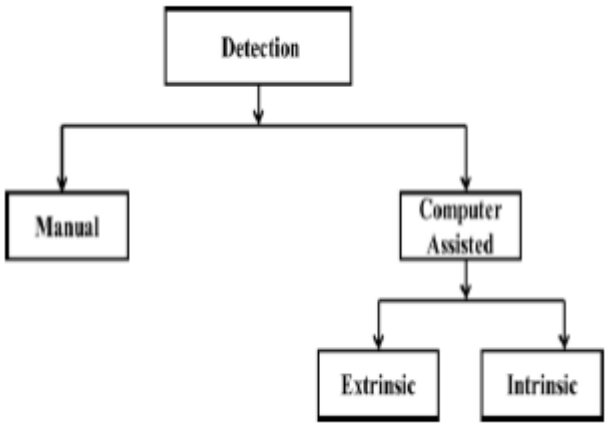


Figure 2

Manual detection is nearly impossible in today's world due to the large size of documents. By using a software or some computer assistance, the process becomes easy. *Extrinsic* plagiarism detection uses one or more source documents to detect plagiarism i.e., documents are checked against source documents. Whereas *intrinsic* type of detection is done by analysis of the writing style variations.

III. PROPOSED METHODOLOGY

Our system is designed to detect copy paste plagiarism between two text documents. For this, we combined the concept of N-grams and substring matching. N- Grams are a sequence of N numbers of words, characters or even sentences sometimes. If there are say 20 words in a document then there will be 20 pairs of N grams. This concept is mainly used in fingerprinting method for detection where each document to be checked for plagiarism in encrypted using a hash function and a unique digital signature is assigned to it. Usually fingerprint is for a single document. However, for this work, we use it as an N-gram fingerprint or fragment fingerprint.

Main goal of the project is to incorporate small strategies together for getting better results. For this, we added the concept of substring matching to be a little bit more confident about the results. For pattern matching [3] we came across two algorithms The Rabin Karp algorithm [4] and The Knuth-Morris-Patt algorithm [5].

The Rabin Karp algorithm has the advantage of a hashing algorithm during the pre-processing phase known as rolling hash function [4] along with substring matching. The hash of the pattern say h (p) matches with the hash of the source h(s) then checked for string's match. This algorithm has the best-case complexity as O (m+n) and the worst case as O (mn).

The KMP algorithm on the other hand uses the degenerating property of the pattern. When there is a mismatch (after some matches), we already know the previous characters of the

window. We avoid matching those numbers of characters again. This algorithm has the worst-case complexity as O (n). Algorithms were analysed on two criteria as first one being the time for the match and the second one number of matches. The analysis result is shown using figures 3 and 4

```
RABIN KARP Test

Enter Text

She sells sea shells on the sea shore

Enter Pattern

sea
Pattern found at index 10
Pattern found at index 28
Took 636953 ns
```

Figure 3. Rabin Karp algorithm test output

```
Knuth Morris Pratt Test

Enter Text

She sells sea shells on the sea shore

Enter Pattern

sea

Match found at index 10
Took 507751 ns
```

Figure 4.KMP algorithm test output

Rabin Karp with its phenomenal ability to find multiple pattern was advantages for detecting plagiarism, despite that KMP is faster.

III. WORKING

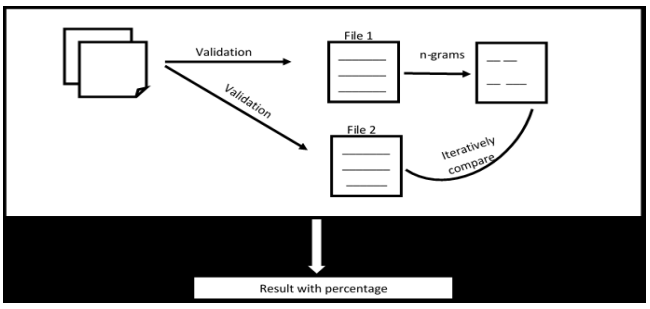
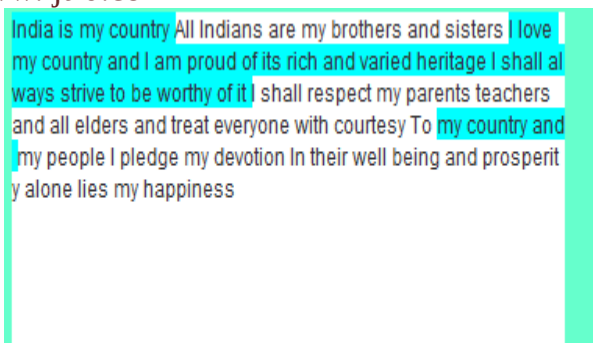


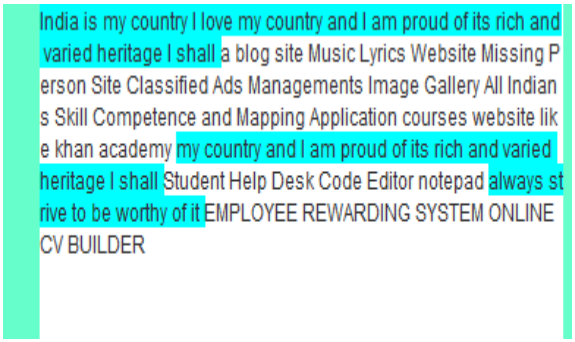
Figure 5.system flow

First the documents are validated by removing punctuations, special and numeric characters. N-grams for first file is pre-processed and validated for redundancy and are stored. All the n-grams are iteratively searched for occurrence in the second file and the match variable is incremented for each occurrence. The percentage of plagiarism calculated as

Percentage= number of matched n-grams / Total number of n-grams

The plagiarized parts are then highlighted in the output





CONCLUSION

This application detects the copy paste text plagiarism between two files with the concept of N-grams with string matching. The plagiarism among the student's assignments. Unlike other plagiarism detection tools [6] PlagScan, ithenticate etc. Which are either premium or has a limit on the usage over tool overcomes these issues and helps n finding plagiarism.

Future Enhancements

To make advancement in the system, the features such as multiple file and global plagiarism checking using different and efficient concepts would be worked upon. Along with this incorporation of databases can be done to deal with large number of files.

References

[1] NamOh Kang and SangYong Han, (2006): "Document Copy Detection System Based on Plagiarism Patterns", A. Gelbukh (Ed.): CICLing 2006, LNCS3878, pp. 571 U 574, Springer-Verlag Berlin Heidelberg 2006.

[2] http://ijsetr.org/wp-content/uploads/2015/12/IJSETR-VOL-4-ISSUE-12-4250-4254.pdf

[3] http://www.student.montefiore.ulg.ac.be/~s091678/files/OHJ2906_Project.pdf

[4] https://www.ijarcsse.com/docs/papers/Volume_4/3_March2014/V4I3-0221.pdf

[5] http://www.geeksforgeeks.org/searching-for-patterns-set-2-kmp-algorithm

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