**MINOR PROJECT-1**

**SYNOPSIS REPORT**

**ON**

**Implementation of Different K-Gram based algorithms for Plagiarism Detection**

**Submitted By**

|  |  |  |
| --- | --- | --- |
| **Name** | **Roll No** | **Branch** |
| Krishna | R110218082 | CSE CCVT |
| Rishab Luthra | R110218122 | CSE CCVT |
| Sanad Kadu | R110218132 | CSE CCVT |
| Vaibhav Mahindru | R110218181 | CSE CCVT |

**Under the guidance of**

Mr. Sandeep Pratap Singh

Department of Virtualization



**SCHOOL OF COMPUTER SCIENCE**

**UNIVERSITY OF PETROLEUM & ENERGY STUDIES**

**Bidholi Campus, Energy Acres, Dehradun – 248007**

 **School of Computer Science**

**University of Petroleum & Energy Studies, Dehradun**

**Project Proposal Approval Form (2020)**

**Minor 1**

**PROJECT TITLE**

Implementation of Different K-Gram based algorithms for Plagiarism Detection

**ABSTRACT**

Plagiarism is considered as the fastest way to accomplish the tasks. According to a survey, 23.2% respondents have been found committing an act of plagiarism. This project aims at implementing a research paper titled "Implementation of Different Algorithm Based K-Gram to Identify Plagiarism on Text-Based Documents". In this paper the author tried to build a plagiarism detection system with Winnowing algorithm as document similarity search algorithm.. The algorithm successfully detects plagiarism.

**INTRODUCTION**

Plagiarism is the wrongful practice of claiming someone else’s ideas, work, findings, in whole or in part as own’s. It is illegal and is subject to academic censure.

The documents that being tested are Indonesian journals with extension .doc, .docx, and/or .txt. Similarity calculation process through two stages, the first is the process of making a document fingerprint using Winnowing algorithm and the second is using Jacquard coefficient similarity.

Contents with less than equal to 30% falls into light plagiarism, moderate plagiarism has similar contents from 30% up to 70%, and more than 70% falls into the category of severe plagiarism. The plagiarized content amount is determined on the basis of similarity scores which is calculated using the Winnowing algorithm.

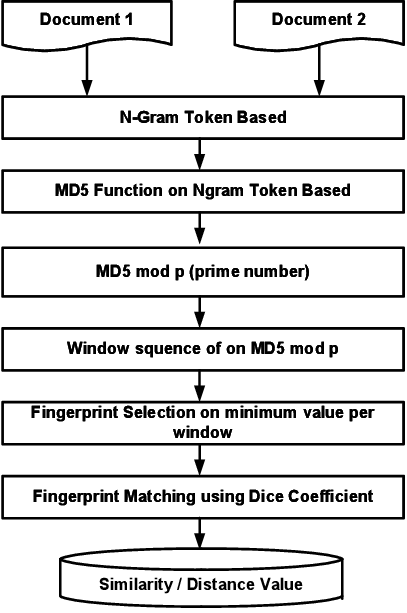
Winnowing algorithm is one of the many Document Fingerprinting methods that is used to detect document similarity by using hashing technique. The hashing technique used is the Rolling Hash and the advantage of the Rolling Hash technique is that the hash value of each character does not need to be recalculated again. By use of the Winnowing algorithm, even smaller parts of the document can be detected for similarities.

In this project we will be using the following algorithms:-

1. Winnowing Algorithm
2. KMP (Knuth Morris Pratt) Pattern Searching Algorithm
3. Rabin-Karp Algorithm
4. Naive Pattern Searching Algorithm

**Winnowing algorithm**

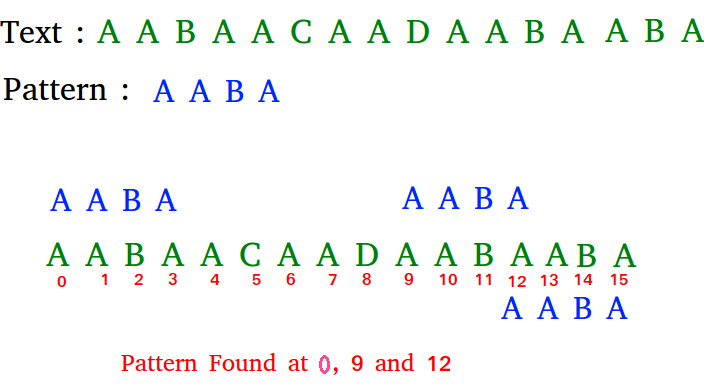
It is done to remove the irrelevant characters such as punctuation and whitespaces. It also converts the capital letters into small letters. Then the preprocessed text is formed into k-grams which is further hashed using the hashing technique. After that, the hashed values are grouped together to form windows from which the smallest values are chosen. The right most value is chosen in case there are more than one smallest value in a single window. Then finally fingerprint is generated using Winnowing algorithm.



**KMP (Knuth Morris Pratt) Pattern Searching**

The basic idea behind KMP’s algorithm is: whenever we detect a mismatch (after some matches), we already know some of the characters in the text of the next window. We take advantage of this information to avoid matching the characters that we know will anyway match.

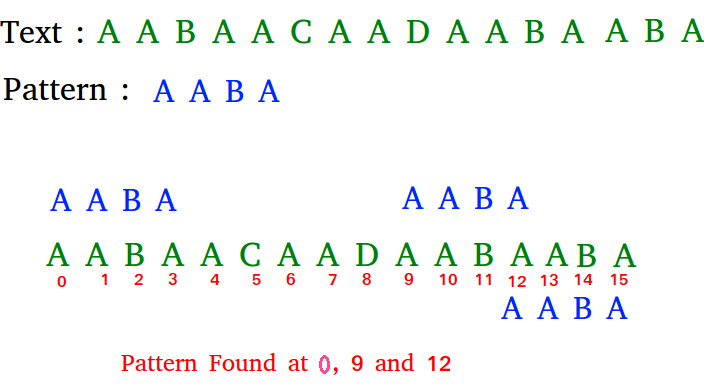
* It preprocesses pattern array and constructs an auxiliary array lps[] which indicates ‘longest proper prefix which is also the suffix’. It is used to skip characters while matching.
* Then lps[] is searched in sub-patterns. More clearly we focus on sub-strings of patterns that are either prefix and suffix.
* For each sub-pattern pat[0..i] where i = 0 to m-1, lps[i] stores length of the maximum matching proper prefix which is also a suffix of the sub-pattern pat[0..i].



**Rabin-Karp Algorithm**

Rabin-Karp algorithm uses hashing and a hash function is calculated for each position. It uses the previous hash to compute next hash, making the process of computing the hash function of adjacent substrings faster.

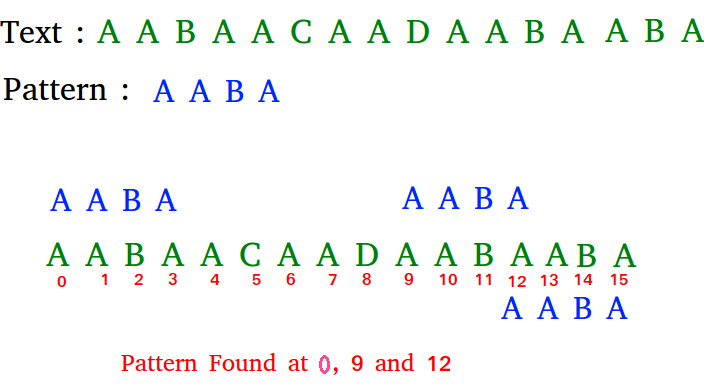
* Choose table size at random to be huge prime.
* Expected running time is O(M + N).
* Θ(MN) worst-case.



**Naive Pattern Searching:**

Slide the pattern over text one by one and check for a match. If a match is found, then slides by 1 again to check for subsequent matches. It does not need any pre-processing phases. We can find substring by checking once for the string. It also does not occupy extra space to perform the operation.

* The time complexity of Naïve Pattern Search method is O(m\*n). The m is the size of pattern and n is the size of the main string
* Worst case is O(m\*(m-n+1))



**PROBLEM STATEMENT**

To implement and compare Winnowing algorithm, Knuth-Morris-Pratt (KMP) algorithm, Rabin-Karp, Naïve string-search algorithm in different test cases for plagiarism detection under the guidance of the research paper, "Implementation of Winnowing Algorithm Based K-Gram to Identify Plagiarism on File Text-Based Document”. The main objective of this project is to determine the level of plagiarism.

**LITERATURE OVERVIEW**

The nomenclature reviewed in this section are: plagiarism, Winnowing algorithm, Knuth-Morris-Pratt(KMP) algorithm and Rabin-Karp algorithm.

Plagiarism originates from the Latin language, 'plagiarius' signifies kidnapper and ‘plagium‘ which means kidnapping. In academics, plagiarism is the act of copying someone’s work and representing as one's own work without adequate acknowledgement.

Winnowing algorithm is a fingerprinting algorithms that can be used for plagiarism detection. It has the advantage of disposing of irrelevant characters such as punctuation and whitespace insensitivity.

KMP is a string matching algorithm which searches a word within a main string. It uses degenerating property hence improving the worst case complexity due to the fact that after encountering a mismatch, it bypasses the re-examination of previously matched characters.

Rabin-Karp is also a string searching algorithm which uses hashing to match the pattern. The use of hash function speeds up the testing of equality of the pattern to the substrings in the text.

Naive Pattern Searching uses a procedure to slide the pattern over text one by one and check for a match. If a match is found, then slides by 1 again to check for subsequent matches. It does not need any pre-processing phases. We can find substring by checking once for the string. It also does not occupy extra space to perform the operation.

|  |  |  |  |
| --- | --- | --- | --- |
| **TITLE** | **LINK** | **AUTHOR** | **REMARKS** |
| In-depth research and working of the algorithms. | https://theory.stanford.edu/~aiken/publications/papers/sigmod03.pdf | Saul Schleimer, Daniel S. Wilkerson, Alex Aiken | This paper explains the need of a plagiarism checking algorithms and explains different algorithms with their working and pseudo codes. |
| To understand the issues of plagiarism and to understand the efficiency of different plagiarism checking algorithms. | https://www.semanticscholar.org/paper/Open-Learning%2C-the-Issue-of-Plagiarism-Efficient-Jazyah/fc7c2087b904989fb49a05a8954b99f305d98c5b?p2df | Yahia Hasan Jazyah | This provides us with the issues caused by plagiarism and about how to fing the most efficient algorithm for the same cause. |
| Implementation of the algorithms and turning pseudo codes into programmable codes | geeksforgeeks |  | It helped us understand the implementation of the algorithms and mould that according to our needs. It also gave us an insight to the different algorithms. |

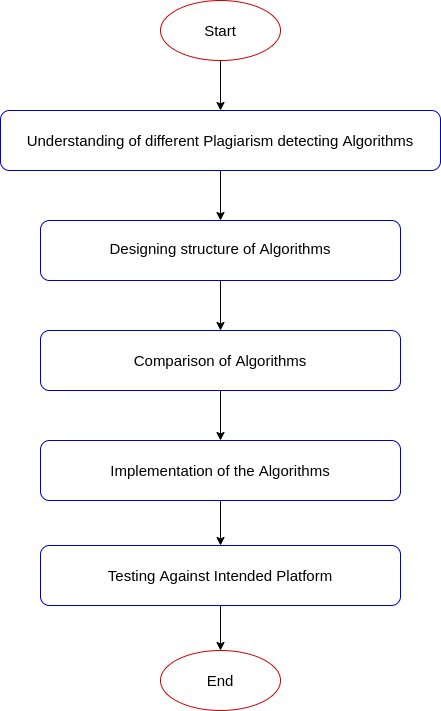
**OBJECTIVES**

1. To implement Winnowing algorithm, Naive Pattern Searching, KMP algorithm and Rabin-Karp algorithm.
2. Comparison of different algorithms based on their space-time complexity.
3. To implement the research paper titled "Implementation of Winnowing Algorithm Based K-Gram to Identify Plagiarism on File Text-Based Document".
4. To determine the percentage of plagiarized content obtained from different algorithms in different scenarios.
5. To the determine the most suitable algorithm mentioned above on the basis of the comparison.

**METHODOLOGY**

The entire implementation of this project can be summarized into the following steps:

1. Understanding of different Plagiarism checking Algorithms: - The plagiarism check algorithms check the similarity between two documents. The algorithms are defined for various types of documents. The most important algorithm for plagiarism checking are:
2. Winnowing Algorithm
3. KMP (Knuth Morris Pratt) Pattern Searching Algorithm
4. Rabin-Karp Algorithm
5. Naive Pattern Searching Algorithm
6. Designing structure of Algorithm: - We will be using the C language to implement all the mentioned algorithms and compare them. Every algorithm has its own unique individuality and space-time complexity, which leads us to our next step.
7. Comparison of Algorithms –Every algorithm has it’s advantages and disadvantages, by comparing all the mentioned algorithms cleanly, fairly and analytically and a well detailed report following it. This will help everyone to understand the algorithms in-depth and will help to choose the most suitable algorithm to use.
8. Implementation of the algorithms: - Here we will implement different algorithms, the algorithms will tell us the plagiarism percentage between two documents.
9. Testing against intended platforms: - In this part we intend to test our final code with respective test cases in form of text files. Testing shall be done in Linux based terminal so as to bring forth errors and unintended functions. Complexity of test cases may be changed to get a better detection of plagiarism percentage.

****

**SYSTEM REQUIREMENT**

**Software requirement**

* Language: C
* Operating System: Windows 7 and above/ any linux distribution
* GCC compiler installed

**Random Access Memory (RAM)**

* 2 GB or above

**Hardware requirement**

* Processor: Intel Core i5-3230 CPU.
* Processor Speed: 2.60 GHz.
* Keyboard Keys: 104 keys.

**PERT CHART**

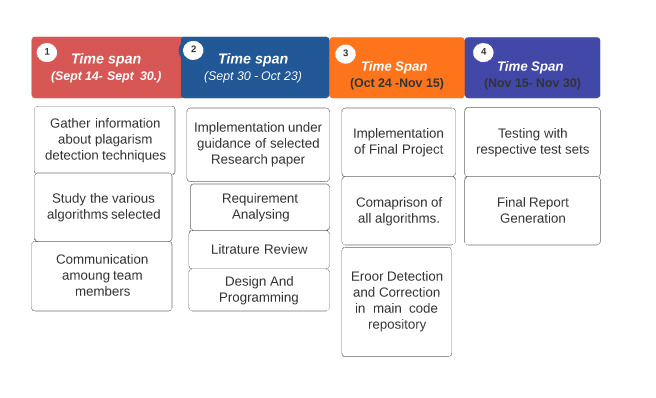


Fig 1: Schedule or Timeline

**REFERENCES**

1. <https://ieeexplore.ieee.org/document/8466429>
2. <https://www.geeksforgeeks.org/rabin-karp-algorithm-for-pattern-searching/>
3. <https://www.geeksforgeeks.org/kmp-algorithm-for-pattern-searching/>
4. <https://theory.stanford.edu/~aiken/publications/papers/sigmod03.pdf>
5. <https://www.geeksforgeeks.org/naive-algorithm-for-pattern-searching/>
6. https://www.semanticscholar.org/paper/Open-Learning%2C-the-Issue-of-Plagiarism-Efficient-Jazyah/fc7c2087b904989fb49a05a8954b99f305d98c5b?p2df

Mentor Approval:-

