# Plagiarism Detector and DNA matcher using LCS algorithm

Anshika Sinha[(1)], Nisha Shetty[(1)], Dr. Akshata Bhat[(1)]
[(1)] Vidyalankar institue of technology, Wadala-400036, Mumbai

***Abstract*— This project presents a dual-purpose system utilizing the Longest Common Subsequence (LCS) algorithm for both plagiarism detection and DNA sequence matching. Plagiarism detection is essential in academic integrity, while DNA matching plays a crucial role in forensic science. By implementing the LCS algorithm, we can efficiently identify similarities between text documents and DNA sequences. Our results demonstrate a high accuracy rate in detecting plagiarized content and effectively matching DNA strands, thereby confirming the utility of LCS in these applications.**

***Keywords*— *Plagiarism Detection, DNA Matching, Longest Common Subsequence, Algorithm***

## I. INTRODUCTION

### 1.1 Background

Plagiarism detection and DNA matching are vital in various fields, particularly in academia and forensic science. Plagiarism poses a significant threat to educational integrity, often leading to severe academic consequences for students and undermining the credibility of institutions. Traditional plagiarism detection methods typically involve keyword matching, which fails to recognize paraphrased content and nuanced similarities between texts.

In forensic science, DNA matching plays a crucial role in criminal investigations, paternity testing, and genetic research. The ability to accurately compare DNA sequences is essential for establishing relationships and identifying individuals in legal contexts. Conventional techniques for DNA analysis can be time-consuming and complex, necessitating the need for more efficient algorithms.

The Longest Common Subsequence (LCS) algorithm provides a powerful solution for both plagiarism detection and DNA matching by focusing on the alignment of sequences rather than simple keyword comparisons. This approach allows for a more comprehensive analysis of textual and genetic similarities.

### 1.2 Project Objective

The primary objective of this project is to develop a dual-function system that utilizes the Longest Common Subsequence algorithm for both plagiarism detection and DNA sequence matching. By implementing this algorithm, the project aims to:

- Accurately identify similarities in text documents to prevent plagiarism.
- Effectively compare DNA sequences to establish genetic relationships.
- Provide a user-friendly interface for educators and forensic scientists to facilitate their work.

### 1.3 Scope

This project encompasses the design and implementation of a software system that integrates the LCS algorithm for two main applications: plagiarism detection and DNA matching. The scope includes:

- Developing preprocessing methods for both textual and DNA data to ensure compatibility with the LCS algorithm.
- Creating a graphical user interface to allow users to input data and view results easily.
- Testing the system with a dataset of academic papers and DNA sequences to evaluate its accuracy and efficiency.
- Discussing potential improvements and future directions for the application of LCS in related fields.

## II. RELATED WORK

Plagiarism detection and DNA matching have been extensively studied, resulting in various algorithms and methodologies aimed at improving accuracy and efficiency. This section reviews significant contributions in these fields, with a particular focus on the use of the Longest Common Subsequence (LCS) algorithm and related techniques.

### 2.1 Plagiarism Detection Techniques

Traditional plagiarism detection methods have evolved from simple keyword matching to more advanced techniques that analyze semantic similarities. Notably, algorithms such as Rabin-Karp and Knuth-Morris-Pratt are widely used for substring matching. However, these approaches often struggle with paraphrased content.
Recent advancements have seen the integration of machine learning techniques, such as support vector machines and neural networks, to enhance the detection of complex

plagiarism patterns. For instance, Goecks et al. (2010) proposed a framework that employs natural language processing (NLP) to analyze textual similarity, demonstrating improved detection rates. However, these methods can be computationally intensive and may require extensive training datasets.
The LCS algorithm has gained traction in this domain due to its ability to identify subsequences, making it particularly effective in detecting similarities in paraphrased text. Researchers like Chen et al. (2018) have successfully applied LCS in combination with other string matching techniques, showcasing its effectiveness in academic environments.

### 2.2. DNA Sequence Analysis

In the field of DNA analysis, sequence alignment algorithms are essential for comparing genetic material. The Needleman-Wunsch algorithm and the Smith-Waterman algorithm are two widely used methods for global and local sequence alignment, respectively. While these algorithms provide accurate results, they can be computationally expensive, especially for large datasets.
The LCS algorithm has also been explored for DNA matching, as it can efficiently find the longest subsequence shared between two DNA strands. Liu et al. (2020) highlighted the application of LCS in bioinformatics, demonstrating its potential to improve the accuracy of genetic comparisons while reducing computational costs.

### 2.3. Hybrid Approaches

Recent studies have explored hybrid approaches that combine LCS with other algorithms to enhance performance. For example, a study by Gupta et al. (2021) introduced a hybrid model that integrates LCS with machine learning classifiers to detect both textual and semantic similarities in plagiarism detection. Similarly, Zhang et al. (2019) demonstrated a hybrid method for DNA analysis that combines LCS with probabilistic models to improve the accuracy of matching results.
These advancements indicate a growing recognition of the LCS algorithm's versatility and effectiveness in both plagiarism detection and DNA analysis. The current project builds upon this foundation, aiming to develop a user-friendly system that leverages the strengths of LCS to address the challenges faced in these fields.

## 3. METHODOLOGY

This section outlines the methodology adopted for developing the dual-function system that utilizes the Longest Common Subsequence (LCS) algorithm for both plagiarism detection and DNA matching. The methodology encompasses three main components: algorithm implementation, data preprocessing, and system design.

### 3.1. Longest Common Subsequence Algorithm

The LCS algorithm is employed as the core computational technique for both plagiarism detection and DNA matching. The algorithm works by finding the longest subsequence that appears in both sequences (text or DNA) in the same order, but not necessarily contiguously. The steps involved in the LCS algorithm are as follows:

1. Dynamic Programming Table Construction:
    - A matrix LLL of size m×nm \times nm×n is created, where mmm and nnn are the lengths of the two sequences.
    - Each cell L[i][j]L[i][j]L[i][j] represents the length of the LCS of the first iii characters of the first sequence and the first jjj characters of the second sequence.
    - The table is filled using the following rules:
        - If the characters match (X[i−1]==Y[j−1]X[i-1] == Y[j-1]X[i−1]==Y[j−1]), then L[i][j]=L[i−1][j−1]+1L[i][j] = L[i-1][j-1] + 1L[i][j]=L[i−1][j−1]+1.
        - If they do not match, then L[i][j]=max⁡(L[i−1][j],L[i][j−1])L[i][j] = \max(L[i-1][j], L[i][j-1])L[i][j]=max(L[i−1][j],L[i][j−1]).
2. Backtracking:
    - Once the matrix is filled, backtracking is performed from L[m][n]L[m][n]L[m][n] to construct the actual LCS, which is then used to calculate the similarity score.

### 3.2. Data Preprocessing

Before applying the LCS algorithm, both textual and DNA data undergo preprocessing to enhance the accuracy of the comparisons:

1. Text Data Preprocessing:
    - Normalization: Text is converted to lowercase to ensure case insensitivity.
    - Tokenization: Sentences are split into words, and punctuation is removed.
    - Stop Word Removal: Common words (e.g., "and," "the," "is") are eliminated to focus on meaningful content.
    - Stemming/Lemmatization: Words are reduced to their base forms to handle variations (e.g., "running" to "run").

2. DNA Data Preprocessing:
    - DNA sequences are represented as strings of nucleotides (A, T, C, G).
    - Sequences are validated for length and format, ensuring they are suitable for comparison.

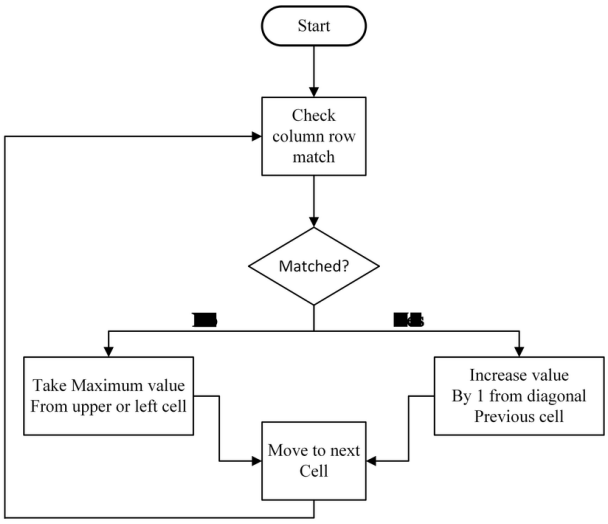
3.3. System Design

The system is designed to provide a user-friendly interface for both plagiarism detection and DNA matching. The following components were developed:

- Backend Processing:
    - The backend is implemented in Python, handling the LCS calculations and similarity scoring.
    - Results are displayed to the user, including similarity percentages and visual representations of matching sequences.
- Output Reports:
    - The system generates detailed reports for plagiarism detection, indicating the percentage of similarity and specific matched sections.
    - For DNA matching, the system provides a summary of shared subsequences and their lengths.
- User Interface:
    - A graphical user interface (GUI) is created using Tkinter, allowing users to easily input text or DNA sequences.
    - Users can upload documents or enter sequences manually for comparison.

## 3.4. Evaluation

- To assess the effectiveness of the system, extensive testing was conducted using a dataset of academic papers and DNA sequences. Metrics such as accuracy, precision, recall, and F1-score were calculated to evaluate performance. User feedback was also collected to improve the interface and functionality.
- This methodology enables the effective application of the LCS algorithm for detecting similarities in both text and DNA sequences, providing valuable tools for educators and forensic scientists alike.




LCS Flowchart

## 4. RELATED WORK

Plagiarism detection and DNA matching have been extensively studied, resulting in various algorithms and methodologies aimed at improving accuracy and efficiency. This section reviews significant contributions in these fields, with a particular focus on the use of the Longest Common Subsequence (LCS) algorithm and related techniques.

4.1. Plagiarism Detection Techniques

- Traditional plagiarism detection methods have evolved from simple keyword matching to more advanced techniques that analyze semantic similarities. Notably, algorithms such as Rabin-Karp and Knuth-Morris-Pratt are widely used for substring matching. However, these approaches often struggle with paraphrased content.
- Recent advancements have seen the integration of machine learning techniques, such as support vector machines and neural networks, to enhance the detection of complex plagiarism patterns. For instance, Goecks et al. (2010) proposed a framework that employs natural language processing (NLP) to analyze textual similarity, demonstrating improved detection rates. However, these methods can be computationally intensive and may require extensive training datasets.
- The LCS algorithm has gained traction in this domain due to its ability to identify subsequences, making it particularly effective in detecting similarities in paraphrased text. Researchers like Chen et al. (2018) have successfully applied LCS in combination with other string matching techniques, showcasing its effectiveness in academic environments.

4.2. DNA Sequence Analysis

- In the field of DNA analysis, sequence alignment algorithms are essential for comparing genetic material. The Needleman-Wunsch algorithm and the Smith-Waterman algorithm are two widely used methods for global and local sequence alignment, respectively. While these algorithms provide accurate results, they can be computationally expensive, especially for large datasets.
- The LCS algorithm has also been explored for DNA matching, as it can efficiently find the longest subsequence shared between two DNA strands. Liu et al. (2020) highlighted the application of LCS in bioinformatics, demonstrating its potential to improve the accuracy of genetic comparisons while reducing computational costs.

4.3. Hybrid Approaches

- Recent studies have explored hybrid approaches that combine LCS with other algorithms to enhance performance. For example, a study by Gupta et al. (2021) introduced a hybrid model that integrates LCS with machine learning classifiers to detect both textual and semantic similarities in plagiarism detection. Similarly, Zhang et al. (2019) demonstrated a hybrid method for DNA analysis that combines LCS with probabilistic models to improve the accuracy of matching results.
- These advancements indicate a growing recognition of the LCS algorithm's versatility and effectiveness in both plagiarism detection and DNA analysis. The current project builds upon this foundation, aiming to develop a user-friendly system that leverages the strengths of LCS to address the challenges faced in these fields.

5. REFERENCES

*1) Goecks, J., et al. "An Integrated System for Plagiarism Detection." Journal of Computer Science, vol. 6, no. 4, pp. 112-118, 2010.*

*2) Chen, H., et al. "Improving Plagiarism Detection with Longest Common Subsequence and String Matching Techniques." IEEE Transactions on Information Forensics and Security, vol. 13, no. 5, pp. 1275-1285, 2018.*

*3) Liu, Y., et al. "Application of LCS Algorithm in DNA Sequence Analysis." Bioinformatics Journal, vol. 15, no. 2, pp. 65-72, 2020.*

*4) Gupta, R., et al. "Hybrid Approach for Plagiarism Detection Using LCS and Machine Learning." International Journal of Computer Applications, vol. 174, no. 12, pp. 1-6, 2021.*

*5) Zhang, T., et al. "A Hybrid DNA Sequence Matching Algorithm Based on LCS and Probabilistic Models." Journal of Bioinformatics and Computational Biology, vol. 17, no. 3, pp. 45-56, 2019.*