Plagiarism Checker Design Document

Mohana Evuri, Shaik Awez Mehtab, Abhineet Majety

Fall 2024

1 Introduction

Concise explanation of the plagiarism checker project, objectives, and scope.

2 Phase 1

2.1 Net Length Calculation

- **Algorithm:** Describe the algorithm used to calculate net length of accurate matches.
- **Function Signatures:**
  - `int calculateNetLength(matches)`

2.2 Longest Approximate Match

- **Algorithm:** Explain the algorithm for finding the longest approximate match and its start indices.
- **Function Signatures:**
  - `Match findLongestApproxMatch(submission, reference)`

2.3 Helpers and Data Structures

- **Helpers:** List helper functions used.
- **Data Structures:** Describe key data structures.
- **Flow:** Overview from `match_submission` call to return.

2.4 Complexity Analysis

- **Time Complexity:** Brief explanation.
- **Space Complexity:** Brief explanation.

3 Phase 2

3.1 Long Pattern Match Detection

- **Algorithm:** Describe detection of long pattern matches against any file.
- **Function Signatures:**

- `bool detectLongPattern(submission, file)`

3.2 Patchwork Plagiarism Detection

- **Algorithm:** Explain detection of patchwork plagiarism.
- **Function Signatures:**
  - `bool detectPatchwork(submission, file)`

3.3 Short Pattern Match Detection

- **Algorithm:** Describe detection of short pattern matches for flagging.
- **Function Signatures:**
  - `bool detectShortPattern(submission, file)`

3.4 Complexity Analysis

- **Time Complexity:** Analysis in terms of number of files (m) and tokens per file (n).
- **Space Complexity:** Analysis in terms of m and n.

3.5 Concurrency Features

- **Threading:** Usage of threading to make `add_submission` non-blocking.
- **Concurrency:** How concurrency is managed in the main thread.

3.6 File Identification

- **Mechanism:** How the set of files to check against is identified within one second after submission timestamp.

3.7 Helpers and Data Structures

- **Helpers:** List helper functions used in Phase 2.
- **Data Structures:** Describe key data structures used.
- **Flow:** Overview from constructor to destructor of `plagiarism_checker_t`.

4 Conclusion

Summary of the design and its effectiveness in detecting plagiarism.