Problem Statement and Goals
Software Engineering

Team 2, SyntaxSentinals
Mohammad Mohsin Khan
Lucas Chen
Dennis Fong
Julian Cecchini
Luigi Quattrociocchi

Table 1: Revision History

Date	Developer(s)	Change
Date1	Name(s)	Description of changes
Date2	Name(s)	Description of changes
...	...	...

1 Problem Statement

[You should check your problem statement with the problem statement checklist. —SS]

[You can change the section headings, as long as you include the required information. —SS]

1.1 Problem

The Measure of Software Similarity algorithm, or Moss algorithm for short, is the current standard for plagiarism detection of code. Broadly speaking, this algorithm works by comparing tokenized code snippets and assigning a similarity score without any weighting based on the complexity of the line being examined. In otherwords, there is an inherent lack of semantic understanding for the code being examined. This gives rise to a major flaw in the Moss algorithm, which is that benign lines of code can be added to a program that do not improve or change functionality but still serve to create an illusion of difference in the eyes of the algorithm. Therefore, even with the Moss algorithm in play, students can easily plagiarize the work of others. Ideally, students should get by on the merit of their own work alone, and a better plagiarism detection can help realize this.

1.2   Inputs and Outputs

The problem will take two or more snippets of code for comparison (n¿=2 code snippets). The desired output is a similarity score between every pairing of the code snippets, and will provide a threshold score to decide whether each score indicates plagiarism or not (n choose 2 scores and 1 threshold for n+1 scores).

[Characterize the problem in terms of "high level" inputs and outputs. Use abstraction so that you can avoid details. —SS]

1.3   Stakeholders

The main stakeholders in this project are professors in any computing and software department, and students enrolled in courses where coding is prevalent. Professors have been identified as stakeholders since they are the people who will be looking out for plagiarism within their own courses. This project provides a tool to give professors the ability to make better predictions on plagiarism. Students have also been identified as stakeholders for two reasons. It would be key to correctly identify the hardwork of a student to prevent others from stealing credit from them, and it would also be critical that a student does not have their hardwork misidentified as another's as it would unjustly punish the original creator. Therefore, the project team must have in mind that we minimize the chance that an innocent student is punished, and maximize the chance that students have their hardwork correctly attributed to themselves alone.

1.4   Environment

This solution will operate on a device, where two files will be fed to a model. The model will leverage hardware provided on the cloud.

[Hardware and software environment —SS]

2   Goals

Goal	Explanation	Reason
item 21	item 22	item 23
item 21	item 22	item 23
item 21	item 22	item 23
item 21	item 22	item 23
item 21	item 22	item 23

- Plagiarism detector will be developed using LLM algorithm

- Plagiarism detector will be comparable to Moss in terms of similarity assignment

- Plagiarism detector will be affected less by dummy lines of code than the Moss algorithm

- Plagiarism detector will be able to compare two or more snippets of code at once (i.e., enough for a classroom)

- Plagiarism detector will complete comparison in reasonable amount of time for use in a classroom.

3   Stretch Goals

- Plagiarism detector is proven to outperform Moss across several test sets.

- Different LLM architectures will be benchmarked against Moss to gauge most optimal archiecture.

- Enhance Moss with our findings to improve on the base algorithm (also necessary if no clear progress can be made in direction of training LLM)

4   Challenge Level and Extras

This project has been assigned a difficulty level of general, and may be subject to change. The aim is to use well known techniques that have been extensively researched, which may push the difficulty to an advanced level, depending on the complexity and feasibility of the research.

The team intends to build an interface to support the project along with a user manual that provides information on how to utilize the interface, for a total of two extras. More ideas for extras can be added in the future.

[State your expected challenge level (advanced, general or basic). The challenge can come through the required domain knowledge, the implementation or something else. Usually the greater the novelty of a project the greater its challenge level. You should include your rationale for the selected level. Approval of the level will be part of the discussion with the instructor for approving the project. The challenge level, with the approval (or request) of the instructor, can be modified over the course of the term. —SS]

[Teams may wish to include extras as either potential bonus grades, or to make up for a less advanced challenge level. Potential extras include usability testing, code walkthroughs, user documentation, formal proof, GenderMag personas, Design Thinking, etc. Normally the maximum number of extras will be two. Approval of the extras will be part of the discussion with the instructor for approving the project. The extras, with the approval (or request) of the instructor, can be modified over the course of the term. —SS]

Appendix — Reflection

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. What went well while writing this deliverable?

2. What pain points did you experience during this deliverable, and how did you resolve them?

3. How did you and your team adjust the scope of your goals to ensure they are suitable for a Capstone project (not overly ambitious but also of appropriate complexity for a senior design project)?