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
Sept 23	Dennis Fong	Add problem statement
Sept 24	Julian Cecchini	Revise problem statement and add goals

1 Problem Statement

Plagiarism detection in code today is not sufficient. There are too many workarounds to be able to avoid being flagged. With the new advances in natural language and programming language understanding, and these advances are yet to be adapted to plagiarism detection algorithms. Proper plagiarism detection helps uphold the merit ofone's achievements, and the addition of semantic understanding of code can help navigate the thin border between plagiarism or the lack thereof.

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

Input: The problem will take two or more snippets of code for comparison (n = 2 or more code snippets).
Output: 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. May also provide an overall score to indicate if plagariasm is suspected somewhere in the dataset provided. (n choose 2 scores, 1 threshold, and 1 overall score for (n C 2) +2 scores).

1.3 Stakeholders

The primary 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. Another stakeholder would be students 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. Lastly, an additional stakeholder could be administrative bodies of schools who would care to incorporate/regulate the use of this detector in their faculty, or give lessens/awareness about it within an official capacity (i.e., meetings or training sessions)

1.4 Environment

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

2 Goals

Goal	Explanation	Reason
Ease of Use	Detector has an intuitive way to insert data and obtain results	This application is expected to be used as a secondary tool for teachers/professors when administering assignments. It should not require in-depth learning, or it will be too inconvenient as an assistant tool for detecting plagiarism. (Measured by actions to complete analysis)
Clarity of Output	Detector explains how to interpret outputs clearly, leaving no ambiguity in whether plagiarism is suspected	If the user does not comprehend the output, it may result in unjust accusations or undetected plagiarism. (Measured by lines of description or number of users who correctly interpret output)
Real-Time Processing	The detector computes results on a dataset of code snippets quickly, enabling professors to incorporate them into evaluations	Since multiple assignments are administered over several weeks, the detector must be fast enough to be realistic for daily use. (Measured by execution time)
False Positive Accuracy	The detector prioritizes minimizing false positives over false negatives	In this case, a false positive could cause harm to an innocent student, while a false negative allows a violation to go unnoticed. The focus is on protecting innocent students. (Measured by false positives and negatives using recall, precision, etc.)
Ethically Sourced Data	The detector uses only data that openly discloses its origins and all data used for the detector is stated for all to see.	In modern ML development, it has become a hot topic for how models get their data. This is because many modern models have used datasets that contain information that was taken from individuals without consent (such as pieces of art). It is important to our team to make it clear this is not precedent and that individuals should have clear consent in being used for training models. (Measured by having accreditation for datasets involved in training)

3 Stretch Goals

Stretch Goal	Explanation	Reason
Online Learning	Provide ability for user to train model on their own datasets.	Without a sufficiently large amount of data to train on (more than will be seen over the duration of 8 months), the model will be biased to a degree and not as widely applicable to different sets of code. If the detector is given the ability to be trained by the user, they can better customize it for their own needs (measure is whether or not the user can change conduct training)
Language Agnostic	The detector can analyze code from a multitude of languages	Having a detector that can draw patterns across different languages will make it adoptable by a wider set of professors who may conduct courses in less popular languages that our detector may have not dealt with at all before. If the detector is restrained to languages such as python or java, we will alienate some of our primary stakeholders.

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 construct a user manual that provides information on how to utilize the detector and to benchmark the performance of different LLM architectures against each other alongside Moss, for a total of two extras. More ideas for extras may be added in the future.

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?

   During this deliverable, everything went smoothly for the most part. Members were assigned different parts, and after everyone had a solid understanding of what our project is, completing our assigned sections had little to no conflict. There was good team communication to start off the semester with.

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

   The main paint points came from gaining a grasp of the project and its scope along with ascertaining what could be goals. The project itself is not simple by any standards. Figuring out what we are doing and ensuring everyone on the team had the same understanding was a challenge. This was resolved by talking the goals over with professor Smith on top of a discussion and thought experiments of what we thought we could get done based on previous work we have done. Aside from that, there were no major paint points.

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)?

   At times, the project felt overly ambitious. However, after thorough review and research, the feasibility and difficulty of the project seemed much more manageable, and not overly ambitious. We ensured the scope of our goals were suitable for capstone by assessing how reachable we felt they were in our own hands. If we felt they were possible but not guaranteed, we kept them as is. If we felt they were completely unreachable or were too simple/arbitrary, we changed them. The goals we ended up with were reasonable, measurable, and, in our opinion, obtainable but not a given, so we must strive hard towards them.