Team Contributions: POC Software Engineering

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

This document summarizes the contributions of each team member up to the POC Demo. The time period of interest is the time between the beginning of the term and the POC demo.

1 Demo Plans

In our proof of concept demonstration, our team will showcase our approach to code plagiarism detection using NLP by comparing two code samples and generating a similarity score.

In our demonstration, we will invoke a command line application. The application will take in two source code files in either Python or Java as input and will return a similarity score as a number from 0.0 to 1.0. A score of 0.0 indicates that the source files have no semantic similarity, and a score of 1.0 means that the two source files are entirely semantically similar. We will invoke this application twice, first with two source files that do obviously different things. These files will have a low similarity score, close to 0.0. The second invocation will be with two source files that do the same thing but differ syntactically. These files will have a high similarity score, close to 1.0. The produced similarity scores should make sense when validated by us as human reviewers.

This demonstration aims to prove that the similarity score produced by our NLP model is a good metric for determining if two pieces of code are plagiarised or not.

2 Team Meeting Attendance

Student	Meetings
Total	3
Mohammad Mohsin Khan	3
Lucas Chen	3
Dennis Fong	3
Julian Cecchini	3
Luigi Quattrociocchi	3

Our team has met on more than just 3 occasions, however, we have only had this many official meetings which we recorded. Much of the time we work together and brainstorm ideas is not recorded explicitly.

3 Supervisor/Stakeholder Meeting Attendance

Student	Meetings
Total	2
Mohammad Mohsin Khan	0
Lucas Chen	0
Dennis Fong	1
Julian Cecchini	2
Luigi Quattrociocchi	0

The two meetings consisted of a supervisor meeting and stakeholder meeting.

Our supervisor, Dr. Hassan Ashtiani, agreed to supervise our project on special terms with his fairly restrictive schedule, which Dr. Smith is aware of. After an initial meeting with him for project scope, our group decided it would be wiser to have our next meeting with him for the POC phase seeing his expertise in ML implementation would be more helpful for starting to code instead of refactoring requirements. Therefore, our next meeting with him has not happened yet but will be happening very soon. The initial scope meeting only had part of the group that held experience with ML algorithms and techniques to better judge with Hassan where our domain knowledge was lacking and what parts of the project we would need to rely on his knowledge for.

The initial skakeholder meeting felt fairly encompassing of all our concerns/ questions. It was held with Dr. Sebastian Mosser, a member of the CAS department who has explored plagiarism detection and NLPs for use in his own classroom before. He made clear difficulties he faced with it and ways we could potentially do better with our own. He also expressed interest in the potential of the project and made clear it could hold use for other professors. Therefore, we did not feel inclined to hold another stakeholder meeting until we have usability or requirement reviews for later SRS revisions or code implementations where we would like to affirm if our direction is still correct. The meeting was held only between Dr. Mosser and a member of the group who had previously TA'd for him as they were able to reach out to Mosser and leverage their familiarity to arrange a meeting scenario Mosser was happy with.

4 Lecture Attendance

Student	**Lectures**
Total	12
Mohammad Mohsin Khan	5
Lucas Chen	5
Dennis Fong	5
Julian Cecchini	5
Luigi Quattrociocchi	5

Every team member has attended every lecture since we started tracking this metric by recording lecture attendance. In the above table, we each have only attended 5 lectures because we didn't track this until a couple weeks into the semester (September 17th is our first recorded lecture). In fact, all team members have attended every lecture, except for the VnV plan lecture on October 23rd, which no one attended because we all had a midterm on that day.

5 TA Document Discussion Attendance

Student	**Lectures**
Total	3
Mohammad Mohsin Khan	3
Lucas Chen	3
Dennis Fong	3
Julian Cecchini	3
Luigi Quattrociocchi	3

6 Commits

Student	**Commits**	**Percent**
Total	130	100%
Mohammad Mohsin Khan	20	15.4%
Lucas Chen	36	27.7%
Dennis Fong	15	11.5%
Julian Cecchini	18	13.8%
Luigi Quattrociocchi	41	31.5%

Part of the reason for Luigi's high number of commits is due to him fixing minor issues (chores) such as moving files or changing issue templates. He was also committing to other people's branches to resolve merge conflicts on their behalf, and leaving review comments with suggested changes, which makes him a co-author of their commit. Every member co-authored commits to some extent, which is why the total number of commits is approximately 30% higher than the number of commits to the main branch.

7 Issue Tracker

Student	**Authored (O+C)**	**Assigned (C only)**
Mohammad Mohsin Khan	6	40
Lucas Chen	83	29
Dennis Fong	0	21
Julian Cecchini	2	20
Luigi Quattrociocchi	5	31

Until now, Lucas has been taking the initiative to create GitHub issues for subtasks within each deliverable, and assign them to the rest of the team. This is why he has the majority of authored issues on the repository. The issues created by other team members were either to document meetings or to record lecture attendance.

Further, Mohsin's assigned commits include those for team and TA meetings, which are assigned to him because he is designated to take notes and meeting minutes.

8 CICD

The project will use continuous integration and continuous deployment (CICD) to run tests and deploy the software. The steps in the CICD pipeline are as follows:

1. Developer creates PR and CICD pipeline will trigger phase 0.
2. Phase 0 will run build and check if the build is successful.
3. If the build is successful, reviewer will review the PR.
4. If the reviewer approves the PR, the CICD pipeline will trigger phase 1.
5. Phase 1 will run tests and check if the tests are successful.
6. If the tests are successful, the CICD pipeline will trigger phase 2.
7. Phase 2 will merge and deploy the software.