Meeting - 10/11/22

Jincheng He (jinchenh@usc.edu, 213-246-9074)

Program Status

Overall, I started my Ph.D. program in Spring 2020, and this semester (Fall 2022) is my 6th semester.

1. **Coursework: All requirements are met.**

   4-unit 600-level courses: CSCI-610 (Program Analysis), CSCI-662 (NLP), CSCI-670 (Algorithm), CSCI-699 (Research Agenda). The rest units can be filled with 790 DR or 794 dissertations.

2. **TAship: All requirements met.** I TA-ed for 561 (AI), 577A (SE), 510 (SE Economics), 585 (Database) in previous semesters.

3. **Qualifying Committee**: (confirmation needed, and I will send out an email to Lizsl once confirmed) I have contacted the following faculty member, and they are willing to be on my committee:

   - G.J. Halfond
   - Sandeep Gupta (EE)
   - Chao Wang
   - Aiichiro Nakano
   - Neil Siegel (Adding Neil needs additional approval from advisor, as he is not a USC tenure track faculty member.)

Research Status

1. **Directed Research Teams (590)**: The CSSE research portal has still been functioning, helping 3 CSSE Ph.D. students to hire M.S. and B.S. students (as unpaid interns, or for units) for their research, and I am maintaining this portal right now. Lizsl knows more about 590 registering and whether this is under some faculty members' name. And I am right now leading two teams, one for research, while another for maintaining CSSE website.

2. **Research Major Task: Categorize software commits by purpose**, and investigate how different types of commits impact software quality.

   - **Motivation** Developers push their commits for different purposes, and different types of commits impact software and its quality differently. Developing a fine taxonomy for them will make purpose-oriented commit analysis possible, and this can be an important part of developer activity study. (This may still need some more references to be more convincing.)
   - **Manual Classification**: We have a set of 1914 commits manually categorized with a refined taxonomy. For manual classification, we read commit messages and code diffs, as well as cross-validating results within our team. If we want to build an auto-classification model based on deep learning techniques, more manually-tagged data will be needed.
   - **Software Metrics**: To give an example of how we may conduct purpose-oriented commit analysis. We used compilability (better change to whether a revision/commit is buildable, to make it broader), software metrics (from SonarQube, PMD and FindBugs/SpotBugs, etc.) as indications of quality changes. And we are looking for other metrics/assessment of quality. For example, one direction to take is to find new analysis tools, or directly get dependency information by applying program analysis techniques. Besides, some tools also provide architectural metrics for software (for example, CAST software) which worth looking into. (I took Dr. Halfond's program analysis course in spring 2022 for this)

- **Automation**: To enable large-scale analysis, we are trying to build models to automate classification, based on commit messages, which is typical, or code changes/diffs, with the help of code summarization tools. (I took Dr. May's Advanced NLP course in Fall 2021 for this). But so far, we haven't achieved an acceptable accuracy, and we are testing on different solutions to overcome the problem, for example, further refining the categories to make changes more distinguishable.