

## **A Study of Repetitiveness of Code Changes in Software Evolution**

This paper looked at repetitiveness of code changes by grouping up the AST tree before and after a commit to see if it matches any other pair. The authors collected a huge code base of billions lines of code and conducted the analysis. The result shows that repetitiveness is very high at small sizes and is insignificant at larger sizes. The author also showed a way to recommend changes based on the result and achieved a accuracy of 30% for top pick and 35% for top 3 pick.

Things I would like to see discussed:

- More details on what makes a code change. The paper discussed detecting code change by AST. It is possible that a AST pair does not necessarily represent the same thing. How accurate is the model.
- Is Java representative as a whole. According to the description in the paper, some of the changes are bug changes. In theory some are specific to languages. So if studying Java enough to conclude in general?
- The analysis is done at commit level. It is very common to have repeated changed when debugging or writing code. It would be nice to discuss the differences between those and if we could gather data on this what would the result say about it.