Project Report 2

Fault Localization using NLP on Bug Reports

with Kostas Kontogiannis

Gurpreet Singh & Paul Bartlett

Software Developers

Contents

Faulty

Project description

The focus will be to design and develop a system that is capable of processing bug reports and extracting useful information about them, and then using that information to provide the developers useful insight into where the bug may be within a large code base. The goal is to reduce the amount of time a developer will need to reach the correct bug after initially reviewing the bug report.

The system will be developed in such a way that it is easily integrated into a continuous delivery pipeline. The project can be divided into three distinct components.

Data Processing

In order for the entire system to work correctly the core essential data processing and analysis has to be effective in detecting the errors. Therefore, the first step is developing a system that can use NLP to process all the bug reports associated with a project to come up with a list of keywords and process the code base to determine a map of relationships between function calls.

User Interface

The next step in delivering the system to a real user, is developing a front end where a user can input a repository for the system to begin processing. Ideal operation of this tool would occur like other DevOps pipeline tools such as Travis.io where a user can link a repository they own and the tool can push it's results back into the bug report for developers to see. There will not be too many interactions available on the front end other than viewing the results of the system and picking new repositories.

Runtime Processing

Since the front end will be making REST API calls to Github repositories, and we need a way to persist processing while providing consistant feedback to users, there needs to be a backend API service allowing those operations to occur. Another task this portion will be responsible for is handling the flow of information when a new bug report appears. The backend will be responsible for detecting this, starting a new processing task, and posting a "Fault Report" back into the bug discussion.

Roles

Gurpreet Singh

- Lead Architect
- Documenter

Paul Bartlett

- Project Manager
- Lead Requirements Analyst
- Lead Tester & Quality Controller

Revised System Requirements

Section A : Data Processing

- **Feature 1:** Able to generate entity relationship rsf from codebase
  - FR 1: Pass code through cdif2rsf to generate rsf
  - FR 2: Clean up incorrect entity and relationships
  - FR 3: Store in accessible data storage for next step to use
- **Feature 2:** Able to generate set of keywords from bug description
  - FR 1: Compare each token to codebase to find valid functions
  - FR 2: Expand initial token set by a factor of 3
  - FR 3: Use NLP to determine question context
- **Feature 3:** Able to combine keywords and rsf into ranked outcomes
  - FR 1: Run LSI on each token and generate search space for each
  - FR 2: Expand the search space for each result in FR 1

  - FR 3: Find similarities between the initial token expansion and the final set of tokens
  - FR 4: Apply ranking equation from research paper to come up with final outcome

Section B : Front-End User Interface

- **Feature 4:** Doing Doing
  - FR 1: Too much doing doing
  - FR 2: Too much doing doing
  - FR 3: Too much doing doing
- **Feature 5:** Doing Doing
  - FR 1: Too much doing doing
  - FR 2: Too much doing doing
  - FR 3: Too much doing doing
- **Feature 6:** Doing Doing
  - FR 1: Too much doing doing
  - FR 2: Too much doing doing
  - FR 3: Too much doing doing

Section C : Back-End Runtime Processing

- **Feature 7:** Doing Doing
  - FR 1: Too much doing doing
  - FR 2: Too much doing doing
  - FR 3: Too much doing doing
- **Feature 8:** Doing Doing
  - FR 1: Too much doing doing
  - FR 2: Too much doing doing
  - FR 3: Too much doing doing
- **Feature 9:** Doing Doing

- FR 1: Too much doing doing
- FR 2: Too much doing doing
- FR 3: Too much doing doing

Project Deviation

The project plan has completed changed into a new idea. In the first report we thought we were going to be working on a system that uses bug data, test data and other external sources to predict the effect of line changes in a code base. Our new idea outlined in the project description above has a much narrower focus and in our opinions a much greater impact. We are excited to work on this new idea