Replication Utilities

This repository includes the steps and information needed to replicate our study.
1- Detection of multi-language code smells occurrences.
2- Detection of bug fixing and bug inducing commits.
3- Extraction of bug topics.
4- Statistical Analysis performed.

This project aims to investigate the evolution of multi-language design smells and the relation between these smells and software fault-proneness.

1. **Detection of multi-language code smells occurrences**
   Location: Folder Detection Approach

   - **Approach:** Detection Approach
   - **Results**: Results of the detection
   - **Evaluation**: Evaluation of the Approach

Getting Started

Running

Run /MLS SAD/src/mlssad/DetectCodeSmellsAndAntiPatterns.java with the path to a directory or file as the only argument. The program will output a CSV file with the code smells and anti-patterns detected in the input source.

Customizing

Change the parameters in /MLS SAD/rsc/config.properties to adapt to your needs. It is currently configured following the default values as thresholds.

Running the tests

The directory /MLS SAD Tests contains tests for each code smell and anti-pattern individually, and two test suites (Applied with the PilotProject).

The tests require the pilot project for detection of anti-patterns and code smells in multi-language systems:

1- Clone the pilot project (PilotProjectAPCSMLS).
2- Create a junction between the folder MLS SAD Tests/rsc and the pilot project folder. On Windows, assuming that the two projects are in the same folder (otherwise, include their paths):
MKLINK /D /J "MLS-SAD\MLS SAD Tests\rsc" "PilotProjectAPCSMLS"

Dependencies

srcML - A parsing tool to convert code to srcML, a subset of XML
Apache Commons Compress - A library for working with archives

Acknowledgments
Loosely inspired by the SAD tool in Ptidej

2. **Detection of bug fixing and bug inducing commits**
   Location: Folder Detection of Bugs
   Scripts: Contains the script used to extract the bugs
   Results: Contains the results of the bug information with the smells
   Evaluation: Contains the manual evaluation of the detection approach

Getting Started

- Run the script Bug_InducingCommits.py to extract the bug information using Pydriller related to a specific project.
- Manual validation of commit inducing of Pydriller and comparison with szz results.

3. **Extraction of bug topics.**
   Approach: Combination of python script and manual analysis
   Results: Results of the topics and the manual validation

Getting Started

- Input file of the commit messages 'commit-messages.csv'.
- Download Mallet.
- Run the script "LDA_Latest.py" with the input file.
- Manual Validation and attribution of tags and keywords.

4. **Data Analysis**
   src: Contains the scripts used for the data analysis
   Results: Contains the results of the statistical tests to answer our research questions

Getting Started

- Input file and scripts for all the statistical tests applied
- Fisher exact tests (Fisher_test_reports)
- Regression
- Correlation