Кластеризация ошибок в программном коде

Научный руководитель: Брыксин Тимофей

Введение

- Люди совершаю схожие ошибки в схожих задачах
- Конкретному человеку свойственны одни и те же ошибки
- В онлайн курсах люди допускают типовые ошибки
- Есть потребность в инструменте для идентификации этих ошибок и их исправлений

Цели проекта

Целью работы был анализ и кластеризация исправлений решений со Stepic-а

Для это нужно было решить следующие задачи:

- Выбрать тип кластеризатора
- Выделить характерные особенности исправлений
- Задать функцию расстояния
- Подобрать параметры кластеризации для получения оптимальных результатов

Подход к решению задачи

- Смотрим на последнюю неверную и первую верную посылки
- Их разница - исправление какой-то проблемы (или нескольких)

- Формируем обучающую (10000) и тестовую (1000) выборки
- Тестовую размечаем руками
- Запускаем кластеризацию
- Оцениваем результаты его работы на тестовой выборке

Использованные материалы

- Comparisons Between Data Clustering Algorithms
  - The International Arab Journal of Information Technology, 2008
  - Osama Abu Abbas
- Clone Detection Using Abstract Syntax Trees
  - Software Maintenance, 1998. Proceedings., International Conference
  - Ira D. Baxter, Andrew Yahin, Leonardo Moura, Marcelo Sant’Anna, Lorraine Bier
- Understanding Source Code Evolution Using Abstract Syntax Tree Matching
  - Association Computing Machinery New York, USA, 2005
  - Iulian Neamtiu, Jeffrey S. Foster, Michael Hicks

Неудачные попытки

- Кластеризовать по значениям метрик
  - Решения одной и той же задачи близки по метрикам
  - Не используется информация об изменениях
- Кластеризовать по разнице значений метрик
  - Исправления небольшие => метрики почти не меняются
  - Многие преобразования вообще не меняют значения метрик
- Кластеризовать по изменению количества ключевых слов/операторов
  - Не учитывается контекст их использования
  - Перемещение куска кода никак не повлияет на количество ключевых слов/операторов

Изменения AST

- Библиотека GumTree позволяет находить преобразования одного дерева в другое
- Кластеризация по списку изменений
  - Вид изменения + тип вершины
  - Вид изменения + тип вершины + тип родителя
  - Вид изменения + тип вершины + тип родителя + тип деда
  - Вид изменения + тип вершины + тип родителя + типы детей родителя + типы детей + глубина вершины

$$sim(a, b) = \frac{LCS(a, b)}{max(a.len, b.len)}$$

$$sim_{nodes}(v, u) = \alpha \cdot sim(v.children, u.children) + \beta \cdot sim(v.neighbors, u.neighbors) +$$

$$+\gamma \cdot (v.parent = u.parent\ ?\ 1 : 0) + \theta \cdot \frac{min(v.depth, u.depth)}{max(v.depth, u.depth}$$

Использованные технологии

- Библиотека GumTree для выделения атомарных изменений AST
- Реализация алгоритма HAC из библиотеки WEKA
- База данных PostgreSQL
- Библиотека Java Diff Utilities от Google для выделения построчной разницы исходных кодов
- Плагин MetricsReloaded для IDEA для подсчёта различных метрик кода

Результаты

- Второй подход (вид изменения + тип вершины + тип родителя) показал наилучший результат
- Среди кластеров, чей размер достиг 5, адекватных 26/31 = 84%
- Собраны примеры таких типов ошибок:
  - Отсутствие import-ов
  - Некорректное название класса
  - Ошибки с операторами сравнения (<, >, <=, >=)
  - Указание package-а
  - Ошибки в сигнатуре main-а
  - Переполнение int-а
  - ...

```
Cluster #26 [size = 80, unknown = 0] tags: RENAME(79), TYPO(3)
Cluster #0 [size = 61, unknown = 1] tags: IMPORT(58), SCANNER(33), CE(5), TYPO(3), LOGIC(1)
Cluster #149 [size = 21, unknown = 6] tags: TYPO(14), CE(5), OUTPUT(3), RENAME(2), DOUBLE_OUTPUT(1), DEQUE(1)
Cluster #216 [size = 21, unknown = 17] tags: DOUBLE_OUTPUT(2), SPACE(1), OUTPUT(1)
Cluster #365 [size = 18, unknown = 5] tags: CE(11), TYPO(9), SEMICOLON(9), OUTPUT(2), STATIC_MISSED(1)
Cluster #366 [size = 18, unknown = 11] tags: STATIC_MISSED(5), TYPO(2)
Cluster #48 [size = 16, unknown = 0] tags: IMPORT(16), BRACKETS(1), CE(1), TYPO(1), THROWS(1), SCANNER(1)
Cluster #6 [size = 15, unknown = 10] tags: RENAME(5)
Cluster #258 [size = 15, unknown = 9] tags: TYPO(4), TYPE(1), STATIC_MISSED(1)
Cluster #98 [size = 13, unknown = 8] tags: RENAME(4), TYPO(3)
Cluster #85 [size = 12, unknown = 0] tags: TYPO(11), CE(2), OUTPUT(1)
Cluster #82 [size = 10, unknown = 0] tags: PACKAGE(10)
Cluster #131 [size = 10, unknown = 6] tags: WRONG_ORDER(2), IMPORT(1), CE(1), TYPO(1), SEMICOLON(1), DEBUG_OUTPUT(1)
Cluster #544 [size = 10, unknown = 1] tags: RENAME(8), STATIC_MISSES(1)
Cluster #163 [size = 9, unknown = 2] tags: RIGOR(6), LOGIC(1)
Cluster #1118 [size = 9, unknown = 0] tags: TYPO(9), TYPE(3), RENAME(1), CE(1), OUTPUT(1)
Cluster #22 [size = 8, unknown = 3] tags: WRONG_IMPORT(4), CE(1), INITIALIZATION(1)
Cluster #142 [size = 8, unknown = 2] tags: IMPORT(5), TYPO(4)
Cluster #849 [size = 7, unknown = 0] tags: BRACKETS(5), TYPO(5), CE(4), RENAME(1), SEMICOLON(1)
Cluster #69 [size = 6, unknown = 1] tags: CASE_MISSED(4), CE(1)
Cluster #297 [size = 6, unknown = 3] tags: TYPO(2), RENAME(1), CE(1)
Cluster #325 [size = 6, unknown = 6] tags:
Cluster #328 [size = 6, unknown = 5] tags: SCANNER(1)
Cluster #7 [size = 5, unknown = 0] tags: RIGOR(5)
Cluster #223 [size = 5, unknown = 3] tags: TYPO(1), SCANNER(1), TYPE(1)
Cluster #569 [size = 5, unknown = 5] tags:
Cluster #617 [size = 5, unknown = 2] tags: RETURN(2), EXIT(2), WRONG_ORDER(1)
Cluster #797 [size = 5, unknown = 1] tags: LOGIC(2), BRACKETS(1), TYPO(1), CASE_MISSED(1)
Cluster #837 [size = 5, unknown = 0] tags: OVERFLOW(5), TYPE(5)
```

Планы на будущее

- Увеличение размера обучающей выборки
- Продолжение экспериментов с функцией расстояния
- Расширение используемой информации
- Использование полученных кластеров для поиска ошибок

Итоги

- Выбран классификатор
- Выбраны характерные особенности кода
- Выбрана функция расстояния
- Построено приемлемое разбиение выборки на ошибки
- Подтверждены идеи об однотипности ошибок в решениях