

Поиск ошибок в коде при помощи глубокого обучения в IntelliJ IDEA

Тучина Анастасия Игоревна
343 группа

научный руководитель: к. т. н., доцент Т. А. Брыксин

СПбГУ

- Большое количество видов ошибок
- Большое количество инструментов и подходов
- На ранних стадиях разработки при использовании динамически типизированных языков часть ошибок остается незамеченной
- Современные решения:
 - Статические анализаторы кода (Google Error Prone, FindBugs)
 - Анализ графов потока управления
 - Поиск известных паттернов ошибок

DeepBugs — инструмент, который:

- Легко расширять на новые классы ошибок
- Использует максимально возможное количество информации из кода (например, имена переменных)

Michael Pradel and Koushik Sen, 2018, DeepBugs: A Learning Approach to Name-Based Bug Detection. In Proceedings of the ACM on Programming Languages Volume 2 Issue OOPSLA, November 2018

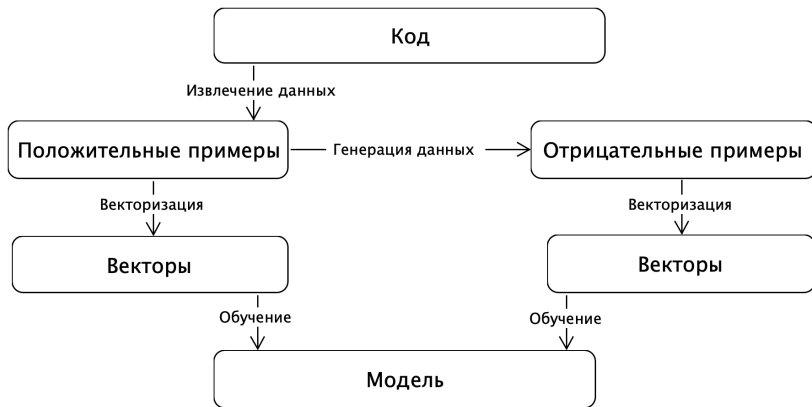
Цель:

- Создание и интеграция детектора ошибок DeepBugs с IntelliJ Platform

Задачи:

- Реализация инспекций кода на JavaScript с использованием предложенных моделей
- Адаптация предложенного подхода для Python с использованием IntelliJ SDK
- Обучение детектора ошибок для каждого из выбранных паттернов
- Реализация инспекций кода на Python, задействующих полученные модели

Архитектура детектора



- Бинарная операция:
(n_{left} , n_{right} , op , t_{left} , t_{right} , k_{parent} , k_{grandP})
- Вызов функции, принимающей два аргумента:
(n_{base} , n_{callee} , n_{arg_1} , n_{arg_2} , t_{arg_1} , t_{arg_2} , n_{param_1} , n_{param_2})

Детектор неправильного порядка аргументов в вызове функции

- Положительный пример:

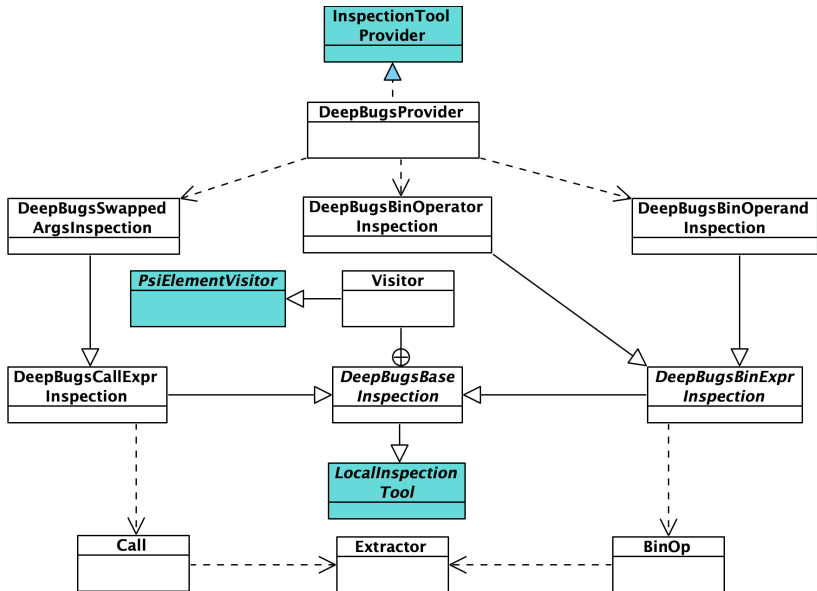
$(n_{base}, n_{callee}, n_{arg_1}, n_{arg_2}, t_{arg_1}, t_{arg_2}, n_{param_1}, n_{param_2})$

- Отрицательный пример:

$(n_{base}, n_{callee}, n_{arg_2}, n_{arg_1}, t_{arg_2}, t_{arg_1}, n_{param_1}, n_{param_2})$

- Каждый из операторов закодирован с помощью one-hot encoding
- Каждый тип данных закодирован с помощью уникального случайного бинарного вектора длины 5
- Каждый тип узла AST закодирован с помощью уникального случайного бинарного вектора длины 8
- Имена идентификаторов векторизуются с помощью Word2Vec. Из векторов составляется словарь. Размер векторного представления – 200.

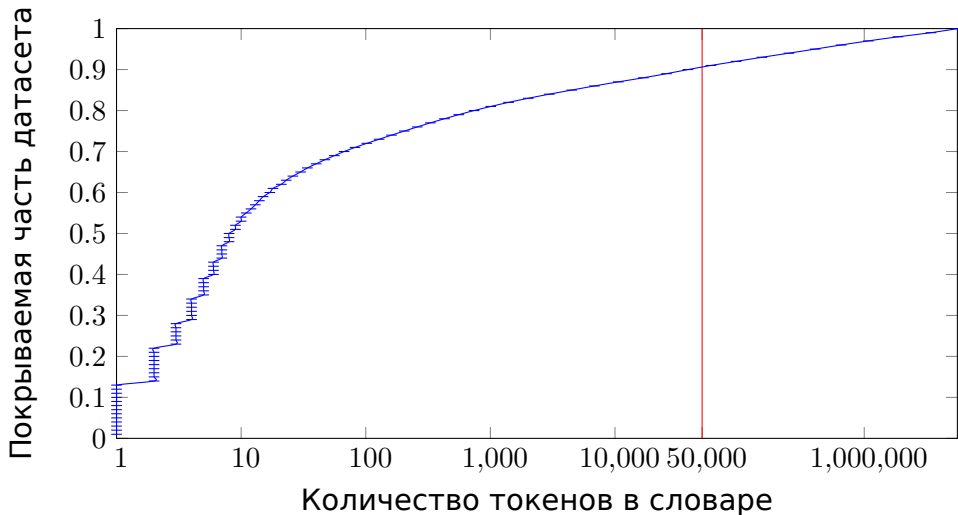
Архитектура плагина



150k Python Dataset¹

- Не содержит дубликатов файлов
- Не содержит обфусцированный код
- Для кода из каждого файла из датасета возможно построить AST с помощью одного из существующих инструментов для Python

¹<https://eth-sri.github.io/py150>



Для обучения было использовано указанное количество фрагментов кода:

Детектор	Количество данных	
	Обучение	Валидация
Детектор неправильных операторов в бинарных выражениях	3677456	1834956
Детектор неправильных операндов в бинарных выражениях	3675316	1815835
Детектор неправильного порядка аргументов в вызовах функций	1181396	568409

Получены модели-детекторы, имеющие следующие точности:

Детектор	Точность
Детектор неправильных операторов в бинарных выражениях	92.08%
Детектор неправильных операндов в бинарных выражениях	85.29%
Детектор неправильного порядка аргументов в вызовах функций	92.06%

Детектор	Всего найдено	Ошибки	Code quality issues	False positives
Детектор неправильных операторов в бинарных выражениях	31	9	14	8
Детектор неправильных операндов в бинарных выражениях	20	5	10	5
Детектор неправильного порядка аргументов в вызовах функций	25	4	15	6
Всего	76	18	39	19

- Реализован набор инструментов для предобработки данных из датасета на Python
- Реализованы плагины для WebStorm и PyCharm, поддерживающие три инспекции кода:
 - Детектор неправильных бинарных операторов
 - Детектор неправильных операндов в бинарных выражениях
 - Детектор неправильного порядка аргументов в вызовах функций (для функций с двумя параметрами)

- Реализация инструментария для предобработки данных:
<https://github.com/ml-in-programming/DeepBugs>
- Реализация плагинов:
<https://github.com/ml-in-programming/DeepBugsPlugin>
- Плагин для PyCharm:
<https://plugins.jetbrains.com/plugin/12218-deepbugspython>
- Плагин для WebStorm:
<https://plugins.jetbrains.com/plugin/12220-deepbugsjavascript>