

Определение авторства разработчиков на основании стиля написания кода

Богомолов Е. О.

Научный руководитель: к.ф.-м.н. Д. Ю. Булычев

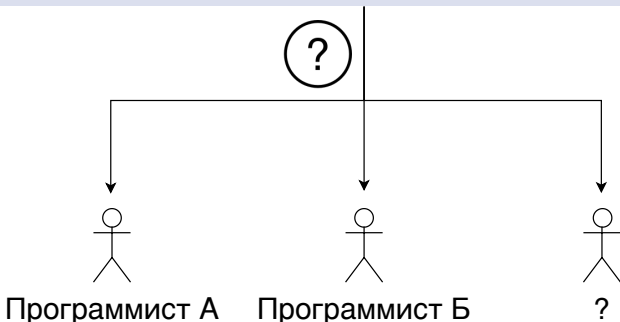
Научный консультант: к.т.н. Т. А. Брыксин

Санкт-Петербургская школа физико-математических и
компьютерных наук

НИУ ВШЭ – Санкт-Петербург

Задача определения авторства

```
int mx=0;  
for(int i=0; i<value.size(); i++){  
    mx=max(values.get(i),mx);  
}  
return mx;
```



- Поиск авторов вредоносного ПО
- Поиск плагиата
- Проверка авторства
- Профилирование программистов

C++¹

- Используются лексические и синтаксические факторы, форматирование
- Обучается случайный лес из 300 деревьев
- Тестирование на решениях 9 задач Google Code Jam, 1600 участников

Python²

- Обучается tree-LSTM по AST и токенам
- Тестирование на решениях 10 задач Google Code Jam, 70 участников

¹A. Caliskan-Islam et al. "De-anonymizing programmers via code stylometry". In: 2015.

²B. Alsulami et al. "Source Code Authorship Attribution Using Long Short-Term Memory Based Networks". In: 2017.

Java³

- Используются лексические, семантические факторы, форматирование
- Обучается модель из трех полносвязных слоев при помощи:
 - Стохастического градиентного спуска
 - Метода роя частиц⁴
- Тестирование на 40 проектах с открытым исходным кодом, у каждого один автор
- Всего 3021 файл, в проектах от 11 до 712

³X. Yang et al. "Authorship attribution of source code by using back propagation neural network based on particle swarm optimization". In: 2017.

⁴J. Kennedy and R. Eberhart. "Particle swarm optimization". In: 1995.

Особенности существующих решений

- Данные существенно отличаются от промышленного кода
 - Специфичные источники данных: соревнования, примеры из книг, домашние задания, проекты с одним автором
 - Число фрагментов доступных для автора в пределах от 9 до 800
- Для различных языков разные решения показывают лучшие результаты
- Лучшие решения не адаптируются для произвольного языка

Цель: создать модель для определения авторства по коду, работающую с произвольными языками программирования и объёмами данных

Задачи:

- Создать инструмент для сбора примеров кода отдельных авторов из произвольных проектов
- Собрать датасеты для тестирования моделей в разных условиях
- Создать прототипы моделей для определения авторства, работающие на основе малого и большого количества данных
- Протестировать модель на доступных датасетах

- Предлагаемый подход работает для произвольных проектов, использующих Git
- История проекта разбивается на коммиты, для них известен автор
- Коммиты разбиваются на изменения методов
- Изменения составляют датасет
- Подход реализован и собраны датасеты на основе IntelliJ IDEA

Сбор данных: датасеты

- IDEA1-4: разный уровень сбалансированности и число авторов
- IDEA4-6: 20 авторов, деление по пакетам разного уровня
- IDEA7: 10 авторов разделение выборок по времени

| | IDEA1 | IDEA2 | IDEA3 | IDEA4 |
|------------------------------|--------------|--------------|--------------|--------------|
| Число авторов | 10 | 16 | 50 | 20 |
| Число примеров (тыс.) | 912 | 648 | 1623 | 1228 |
| Макс. / Мин. примеров | 5 | 3 | 32 | 8.5 |
| Примеров в группе | 1 | 16 | 16 | 16 |

Модель: мало данных

- Случайный лес, число деревьев подобрано экспериментально и равно 500
- В качестве факторов используются частоты путей⁵ и токенов в AST
- Факторы фильтруются на основе совместной информации между фактором и автором
- Для получения токенов и путей используется специально написанный инструмент, поддерживающий популярные языки программирования
- Произведено сравнение с имеющимися решениями

⁵U. Alon et al. "A general path-based representation for predicting program properties". In: 2018.

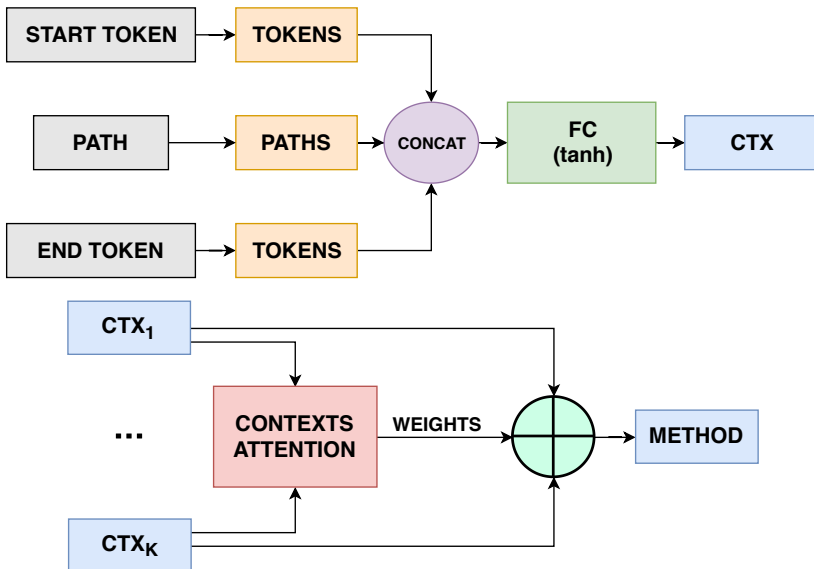
Модель: много данных

- Для работы адаптирована архитектура code2vec⁶
- Модель уже улучшила результаты в определении имени метода и генерации описания по коду
- Архитектура расширена для работы с несколькими фрагментами кода
- Произведено тестирование на данных IntelliJ IDEA и существующих датасетах

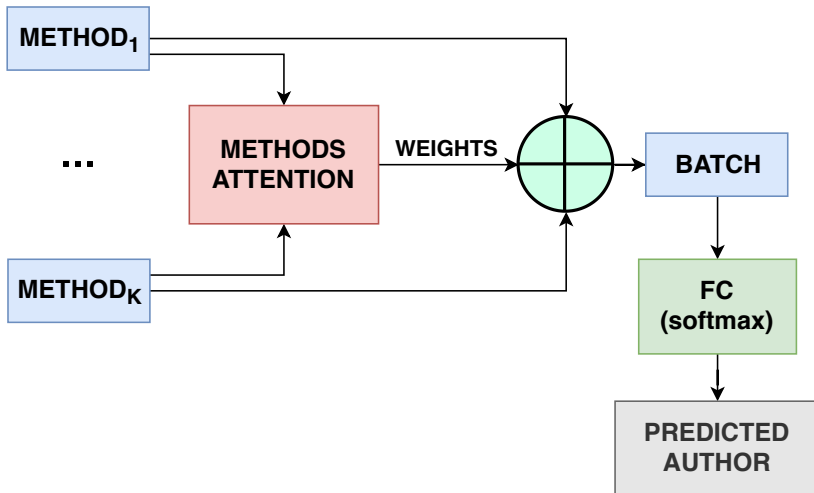
| | IDEA1 | IDEA2 | IDEA3 | IDEA4 | IDEA5 | IDEA6 | IDEA7 |
|------------|-------|-------|-------|-------|-------|-------|-------|
| Acc | 74.4% | 99.7% | 93.5% | 97.8% | 92% | 87.3% | 89.6% |
| MAP | 73.8% | 99.7% | 91.1% | 97.4% | 93% | 85.9% | 88.5% |

⁶U. Alon et al. "code2vec: Learning Distributed Representations of Code". In: 2018.

Модель: архитектура code2vec



Модель: модернизация code2vec



Тестирование на существующих датасетах








- Модель на основе случайного леса повторяет или улучшает результаты для трех языков
- Нейросетевая модель работает хуже из-за маленького размера обучающей выборки

| ЯП Число авторов | C++ 1600 | Python 70 | Java 40 |
|---------------------------|---------------|--------------|------------|
| Caliskan, 2015 | 92.83% | 72.9% | - |
| Alsulami, 2017 | - | 88.86% | - |
| Yang, 2017 | - | - | 91.06% |
| Эта работа, нейросеть | 41.7% | - | 86% |
| Эта работа, случайный лес | 92.7% | 94.1% | 97% |

Результаты

- Реализован инструмент, генерирующий датасеты для определения авторства по проектам, использующим Git
- Создан набор из 7 датасетов изменений методов в IntelliJ IDEA
- Созданы и протестированы модели для работы с большим и ограниченным числом данных
- Модель на основе случайного леса повторяет или улучшает точность определения авторства для Java, Python и C++
- Часть работы представлена на MSR'19⁷

⁷[Vladimir Kovalenko et al. "PathMiner: A Library for Mining of Path-Based Representations of Code". In: 2019.](#)

-  Alon, U. et al. "A general path-based representation for predicting program properties". In: 2018.
-  – . "code2vec: Learning Distributed Representations of Code". In: 2018.
-  Alsulami, B. et al. "Source Code Authorship Attribution Using Long Short-Term Memory Based Networks". In: 2017.
-  Caliskan-Islam, A. et al. "De-anonymizing programmers via code stylometry". In: 2015.
-  Kennedy, J. and R. Eberhart. "Particle swarm optimization". In: 1995.
-  Kovalenko, Vladimir et al. "PathMiner: A Library for Mining of Path-Based Representations of Code". In: 2019.
-  Yang, X. et al. "Authorship attribution of source code by using back propagation neural network based on particle swarm optimization". In: 2017.

Результаты: IDEA

| | Точность | Acc₂ | Acc₅ | MAP |
|--------------|-----------------|------------------------|------------------------|------------|
| IDEA1 | 74.4% | 86.3% | 95.7% | 74.3% |
| IDEA2 | 99.7% | 100% | 100% | 99.7% |
| IDEA3 | 94.2% | 97.2% | 99% | 91.9% |
| IDEA4 | 97.8% | 99.3% | 99.8% | 97.4% |
| IDEA5 | 92% | 96.5% | 99.1% | 93% |
| IDEA6 | 87.3% | 93.2% | 97.7% | 85.9% |
| IDEA7 | 89.6% | 95.6% | 98.9% | 88.5% |

Датасеты IDEA

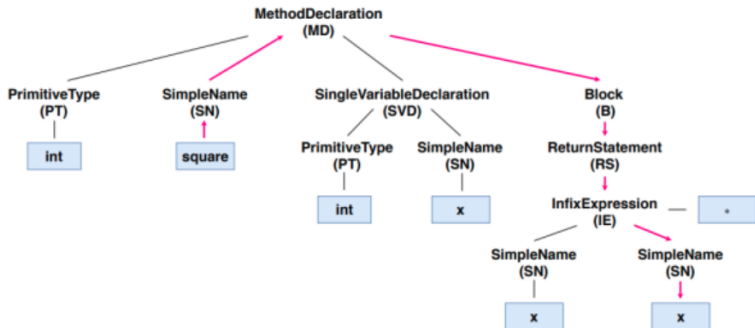
| | IDEA1 | IDEA2 | IDEA3 | IDEA4 |
|------------------------------|-------|-------|-------|-------|
| Число авторов | 10 | 16 | 50 | 20 |
| Число примеров (тыс.) | 912 | 648 | 1623 | 1228 |
| Макс. / Мин. примеров | 5 | 3 | 32 | 8.5 |
| Примеров в группе | 1 | 16 | 16 | 16 |

| | IDEA5 | IDEA6 | IDEA7 |
|------------------------------|-------|-------|-------|
| Число авторов | 20 | 20 | 10 |
| Число примеров (тыс.) | 1228 | 1228 | 912 |
| Макс. / Мин. примеров | 8.5 | 8.5 | 5 |
| Примеров в группе | 16 | 16 | 16 |
| Деление по пакетам | 1 | 2 | 0 |
| Деление по времени | - | - | + |

Абстрактное синтаксическое дерево

```
int square(int x) {  
    return x * x;  
}
```

(a) An example code snippet



(b) The snippet's syntax tree. An example of a path is highlighted in pink.