Совершенствование методов классификации типов ошибок в решениях задач онлайн-курсов по программированию

Автор: Смирнов Олег Евгеньевич, 271 группа
Научный руководитель: к. т. н., доцент Брыксин Т. А.

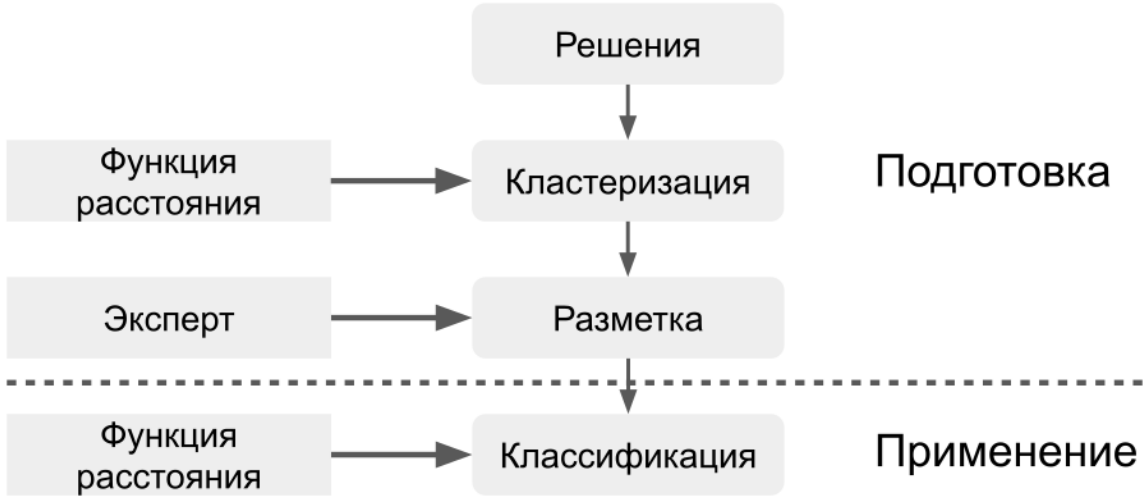
Санкт-Петербургский государственный университет
Кафедра системного программирования

6 июня 2020г.

Актуальность

- Массовые онлайн-курсы по программированию невероятно популярны
- Проблема — получение обратной связи от преподавателя
- Множество способов генерации подсказок, универсального подхода не существует

- В рамках работы исследовательской группы лаборатории *JetBrains Research* была опубликована статья, в которой предлагался новый подход к генерации подсказок:



Исходная схема решения

- Алгоритм показал достойные результаты
- Решено было продолжить работу над его улучшением

Цель и задачи

- **Цель:** совершенствование существующего подхода для определения и классификации типичных ошибок в коде решений задач онлайн-курсов по программированию.
- **Задачи:**
  - Проанализировать подходы машинного обучения для классификации изменений в исходном коде
  - Агрегировать наиболее подходящие и разработать улучшенный алгоритм классификации
  - Исследовать возможность ускорения отдельных частей существующего подхода
  - Провести сравнительный анализ алгоритмов посредством экспериментов с решениями конкретных учебных задач
  - Провести апробацию лучшего подхода с использованием имеющихся данных сразу по нескольким задачам

Аугментация данных

- Набор данных из размеченных сценариев редактирования был сравнительно мал
- Попробовали применить указанную методику расширения датасета
- Сценарии редактирования стали длиннее, и, соответственно, стали больше походить на сценарии из реальной жизни

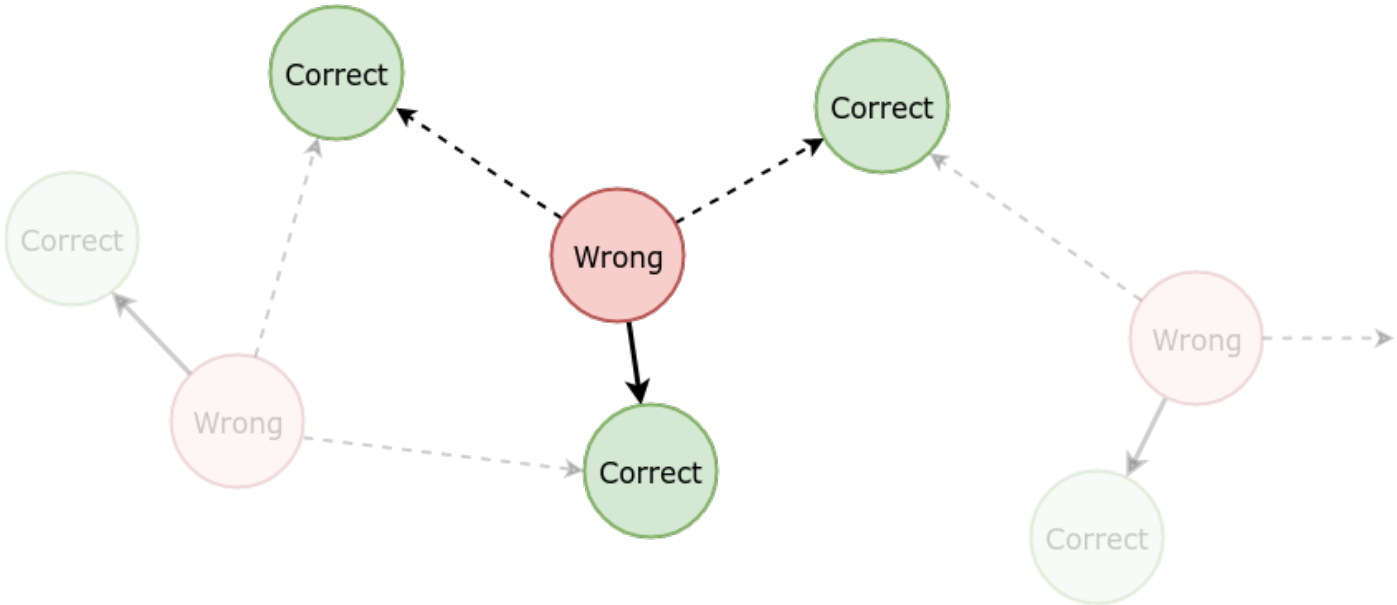
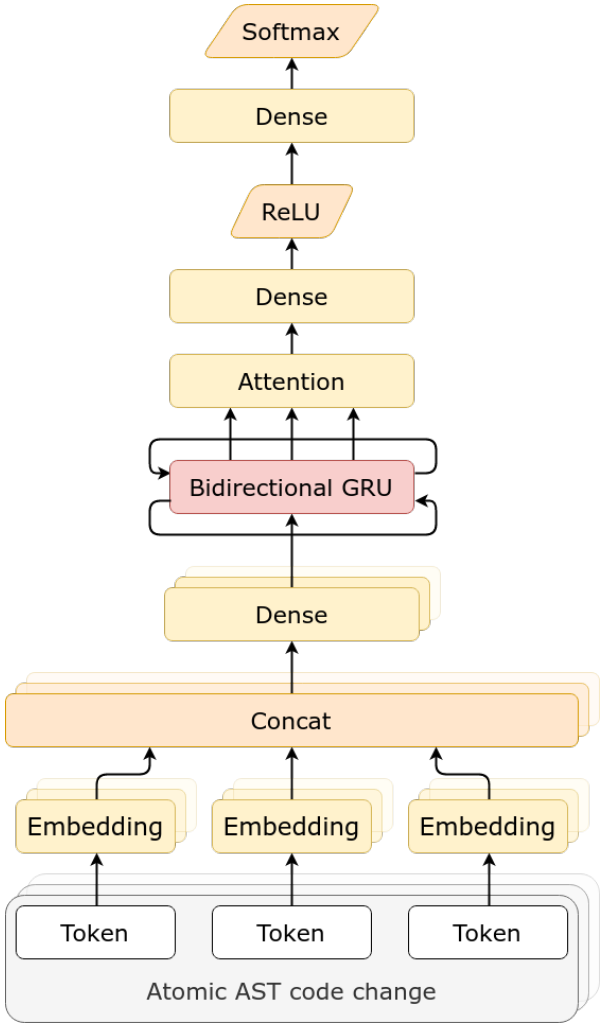


Схема аугментации в пространстве решений по трём соседям

Ускоренный поиск ближайших соседей

- Часть исходного подхода с поиском ближайших правильных решений — самая медленная
- Предлагаемая эвристика позволяет уменьшить асимптотику алгоритма поиска с $\mathcal{O}(n)$ до $\Omega(\sqrt{n})$, где n есть число правильных решений в тренировочной выборке
- Качество изменилось совсем незначительно, время работы подхода уменьшилось почти в 4 раза (см. слайд №10)

Улучшенный классификатор скриптов редактирования



Архитектура глубокой нейронной сети для классификации изменений в коде

Апробация

- Были выбраны 6 задач, отличающихся тематикой и средним размером решения

Problem	N	LOC	Train		Validate	Test
			$N_{correct}$	N_{wrong}	N_{wrong}	N_{wrong}
deserialization	2624	19	2294	1643	60	140
factorial	8883	14	8330	5357	60	140
loggers	3403	15	3137	2067	60	140
reflection	4217	11	3827	2742	60	140
filter	2673	24	2549	1747	60	140
integral	4952	12	4539	2859	60	140

Набор размеченных данных по шести задачам с платформы Stepik

Независимое тестирование классификаторов

- Эксперименты проводились на тестовой выборке из 140 решений
- Гипер-параметры моделей подобраны по валидационной выборке из 70 решений
- PR AUC — Precision-Recall Area Under Curve (чем ближе к 1, тем лучше)

Model	*deserialization*	*factorial*	*loggers*	*reflection*	*filter*	*integral*
KNN-1	0.645	0.681	0.728	0.688	0.357	0.578
KNN-5	0.679	0.726	0.755	0.765	0.388	0.603
KNN-15	0.708	**0.736**	0.783	0.772	0.400	0,613
AttBiGRU	**0.731**	0.717	0.794	**0.780**	0.562	0.624
KNN-embed	0.693	0.666	**0.795**	0.774	**0.588**	**0.634**

Сравнение классификаторов по метрике качества PR AUC

Измерение времени поиска ближайших соседей

- Complete search — поиск по всем правильным решениям
- Heuristic search — поиск только про репрезентативным элементам каждого кластера

Problem	$N_{correct}$	LOC	Complete search		Heuristic search	
			PR AUC	Time (s)	PR AUC	Time (s)
deserialization	2294	19	0.731	453,6±4,9	0.715	101,2±3,8
factorial	8330	14	0.717	209,1±7,7	0.694	42,9±0,5
loggers	3137	15	0.794	208,5±4,3	0.753	84,7±5,5
reflection	3827	11	0.780	157,1±1,9	0.766	54,3±2,5
filter	2549	24	0.562	2029,7±19,5	0.541	172,8±1,2
integral	4539	12	0.624	1097,1±13,8	0.571	121,2±1,6

Замеры времени по избранным задачам для двух алгоритмов

Результаты

- Проведен обзор предметной области, изучены подходы и методы машинного обучения для кластеризации и классификации изменений в коде
- Разработан нейросетевой алгоритм классификации сценариев редактирования кода в учебных задачах
- Реализован эвристический алгоритм поиска в пространстве решений, значительно превосходящий по скорости исходный аналог
- За время работы над курсовой были сделаны два пулл-реквеста в публичный репозиторий проекта[1]
- Проведена апробация нового подхода и сравнительный анализ алгоритмов на реальных данных по шести учебным задачам с платформы Stepik

[1] https://github.com/JetBrains-Research/bugs-classification