

Анализ сложности задач на Stepik по условию и коду решения

Студент: Кузнецов Александр

Научный руководитель: Брыксин Тимофей

Мотивация

- Хотелось бы научиться генерировать задачи по программированию разной сложности
- Для этого нужно сначала научиться по задаче определять ее сложность

Постановка задачи

1. Рассмотреть подходы и алгоритмы, использующиеся для задач классификации
2. Собрать и подготовить входные данные для алгоритмов
3. Испытать алгоритмы на наборе задач и экспериментально получить сложность этих задач
4. Сравнить полученные сложности с заведомо известными сложностями задач из набора
5. Определить, справился ли какой-нибудь алгоритм с задачей

Подходы к решению

- Классификация текста:
 - Pratiksha Y. Pawar and S. H. Gawande, Member, IACSIT, «*A Comparative Study on Different Types of Approaches to Text Categorization*», International Journal of Machine Learning and Computing, Vol. 2, No. 4, August 2012
- Классификация кода:
 - Статический анализ. [Esubalew Alemneh, Abdul Azim Abd Ghani, Rodziah Atan, «*SAT4BSC: A Static Analysis Tool for BPEL Source Codes*», IJCSMC, Vol. 3, Issue. 2, February 2014, pg.659 – 665]
 - Использование нейронных сетей. [Lili Mou, Ge Li, Lu Zhang, Tao Wang, Zhi Jin, «*Convolutional Neural Networks over Tree Structures for Programming Language Processing*», Association for the Advancement of Artificial Intelligence, 2016]

Определение сложности по тексту условия

- Выделение векторов признаков:
 - Bag-of-words
 - Term frequency — inverse document frequency
- Классификаторы:
 - Stochastic Gradient Descent
 - k-Nearest Neighbors
 - Random forest
 - etc

Определение сложности по коду решения

- Рассмотрены два подхода:
 - Каждому узлу AST ставится в соответствие некоторое значение; вектор признаков — список значений, полученных в порядке обхода дерева
 - Рассчитывается несколько метрик для целой программы (цикломатическая сложность, количество операторов и операндов, etc.) [Vera Barstad, Morten Goodwin, Terje Gjørseter, «*Predicting Source Code Quality with Static Analysis and Machine Learning*», NIK-2014]
- Нейросеть включает Dropout + BiLSTM + Dropout
 - Eliyahu Kiperwasser, Yoav Goldberg, «*Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations*», Transactions of the Association for Computational Linguistics, vol. 4, pp. 313–327, 2016

Инструменты

- Классификация текста (выделение векторов признаков (features), тренировка и тестирование) — библиотека sklearn
- Классификация кода:
 - Выделение векторов признаков — вручную
 - Тренировка и тестирование — нейронные сети (библиотека keras)

Наборы данных

- Для текста — около 1900 различных задач с codeforces
 - Априорная сложность задачи определяется ее порядковым номером в соревновании
- Для кода — около 155000 программ на языке Java с платформы Stepik
 - Априорная сложность задачи определяется статистикой ее решения (среднее значение функции от количества попыток и времени, потраченного на решение)

Определение сложности по тексту условия: результат

- Folding для тестирования, n_folds = 8

| | Stochastic Gradient Descent | k Nearest Neighbors | Random Forest | Ridge Classifier | Single-Layer Perceptron | Multi-Layer Perceptron |
|--------|-----------------------------|---------------------|---------------|------------------|-------------------------|------------------------|
| BoW | 72,1 % | 67,2 % | 74,7 % | 75,2 % | 73,2 % | 71,0 % |
| Tf-Idf | 73,2 % | 64,9 % | 76,0 % | 70,0 % | 67,1 % | 59,1 % |

- Проблема: наличие легенд в условиях

Определение сложности по коду решения: результат

| Метод | Точность |
|---|----------|
| Обход AST, метрика — глубина вершины | 81,60 % |
| Обход AST, метрика — #вершин в поддереве/#вершин в поддереве родителя | 88,30 % |
| Метрики программного кода | 87,35 % |
| Комбинация двух предыдущих методов | 88,50 % |
| Удаление выбросов | 92,00 % |

Определение сложности: результат

| Задача | Вердикт для условия | Вердикт для кода |
|--|---------------------|------------------|
| Перевернуть строку, не используя циклы | easy | easy |
| Реализовать Rope | hard* | hard |
| Codeforces Round #456, Div.2, Task D | easy | hard |
| Codeforces Round #455, Div.2, Task E | hard | hard |
| Codeforces Round #464, Div.2, Task B | hard | easy |

Резюме

- Рассмотрены и реализованы различные подходы к построению классификатора текстов условий и кодов решений
- Собраны входные данные
- Измерена точность работы этих подходов на наборах задач
- Результат для текстов условий — не очень хороший
- Результат для кодов решений — приемлемый