N-gram-based Authorship Identification of Source Code

Yunpeng Yang[b], Leilei Kong[a*], Zhongyua Han[a], Yong Han[a] and Haoliang Qi[a]

[a] *Foshan University, Foshan, China*
[b] *Heilongjiang Institute of Technology, Harbin, China*

Abstract

This paper focuses on the task of source code author identification published on PAN@FIRE2020 (Information Retrieval Evaluation Forum) which is to identify the most likely author of the code given a set of C++ source code without defined authors. This research is useful in some cases, such as detecting malware authors, solving academic cheating and online coding competition cheating problems. In the evaluation, we regard the source code author recognition task as a multi-classification task, and use word n-gram and character n-gram to extract features to train a logistic regression classifier. In the final results, the accuracy of our method reached 0.9428, ranking second. The experiments show that using the character n-gram as features is the best way to improve the prediction accuracy.

Keywords 1

N-gram, Authorship Identification, Source Code, Multi-classification, Logistic Regression

1. Introduction

The research on source code author identification is maturing gradually with more of an emphasis on identifying the author of a piece of code. Although the computer language is more standardized than the natural language, in most cases, the programs written by different people will be very different.

Focused on the source code author identification, PAN@FIRE2020 proposed a task named AI-SOCO(Authorship Identification of SOurce COde)[1]. In AI-SOCO, the task of source code author identification is defined as follows: given the pre-defined set of source codes and their writers, the task is to build a system that is able to detect the writer given any new, unseen before source codes from the previously defined writers list.

In this evaluation, we use the method of multi-classification with character-based 2 to 7-gram to realize the task of AI-SOCO. After word segmentation, the method based on TF-IDF is used to filter the ngram features. Finally, the filtered results are used as features to train the classifier. After analyzing the results of many experiments, we choose the logistic regression as classifier from random forest, logistic regression, language model, neural network. The prediction accuracy of this method for AI-SOCO reaches 0.9428, achieving the second place.

2. Methods

Figure 1 depicts our method for AI-SOCO.

Firstly, given a set of training data made up of c++ codes written by different authors, the original codes are dealed with word-based n-gram and character-based n-gram respectively, after analyzing the

Forum for Information Retrieval Evaluation, December 16-20, 2020, Hyderabad, India

EMAIL: yunpengy@outlook.com (A. 1); kongleilei@fosu.edu.cn (A. 2)(*corresponding author); hanzhongyuan@fosu.edu.cn (A. 3); hanyong@fosu.edu.cn (A. 4); qihaoliang@fosu.edu.cn (A. 5)
ORCID: 0000-0002-6247-9999 (A. 1); 0000-0002-4636-3507 (A. 2); 0000-0002-4636-3507 (A. 3)

CEUR Workshop Proceedings (CEUR-WS.org)

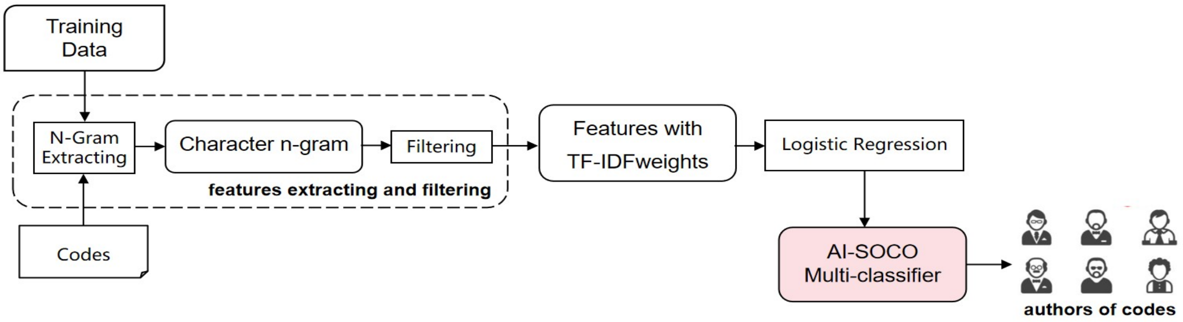

Figure 1: Method for AI-SOCO

detailed data of data set in Table 1 and Table 2. By comparing the two kinds of word segmentation, we find that the features extracted by character 2-gram to 7-gram can obtain better performance. Then, we use TF-IDF method to calculate TF, IDF, DF, TF * IDF values of these features. Especially, the IDF is computed using Eq.1:

$$IDF\ (t) = \log \frac{1 + n_d}{1 + df(d,t)} + 1 \tag{1}$$

For the purpose of feature filtering, we delete some features whose DF values are too high or too small, and regularize the TF * IDF values of the remaining features as new features, denoted as Features with TF-IDF weights in Fig.1. Lastly, the logistic regression is used as the multi-classifier to train the model of AI-SOCO. The parameter of the logistic regression model we use is proposed by sklearn[1], the parameters are set as C=1.0, max_iter=100, multi_class='ovr', penalty='l2', solver='liblinear' and tol=0.0001.

3. Experiments

3.1. Dataset

The dataset is composed of source codes collected from the open submissions in the Codeforces online judge. The total number of source codes in the dataset is 100,000, which are from 1,000 authors respectively. The source codes of each author are 100 and all of them are C + + codes. Detailed information for the dataset is given in Table 1 and Table 2.

3.2. Experimental Results

The performance of source code author identification task is evaluated by accuracy. Table 3 shows the final evaluation results of top 5.

We have tried many feature extraction methods for observing their effects. The experimental results are shown in Table 4 and Table 5.

According to Table 4, we can see that the logistic regression model and the random forest model are better in the word 1-gram. Therefore, we train the two models at the same time, and through the analysis of the results, we choose the logistic regression model.

In conclusion, firstly, using TF-IDF filtered features to train the model can get higher accuracy. Secondly, the effect of character n-gram is better than word n-gram. So in the final evaluation, we use TF-IDF filtered character 2+3+4+5+6+7gram feature to learn the logic regression model, and the accuracy rate is 0.93556 in development set.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
We used the logistic regression model in this website to do multi-classifier tasks

Table 1
The data set used in this evaluation

Category	Number	Category	Number
Users Count	1,000	Unique Problems	6,553
Solutions Count	100,000	Maximum Solutions/Problem	61
Tokens Count	22,795,141	Minimum Solutions/Problem	1
Whitespaces Count	46,944,461	AVG. Solutions/Problem	15.26
Unique Tokens	1,171,991	AVG. Solutions/Codeforces Index	2,439.02
AVG. Solutions/User	100	Median Solutions/Problem	12
AVG. Tokens/Solution	227.951	Unique Countries	78
AVG. Whitespaces/Solution	469.445	AVG. Solutions/Country	1,282.05
Maximum Tokens in a Solution	10,189	Minimum Tokens in a Solution	3

Table 2
Number of Problems pre Language

Language	Problems
GNU C++	10141
GNU C++0x	1584
GNU C++11	23849
GNU C++14	35222
GNU C++17	26091
GNU C++17(64)	869
GNU C++17 Diagnostic	5
MS C++	2005
MS C++2017	234

Table 3
The final evaluation results

Ranking	Team Name	Accuracy
1	UoB	0.9511
2	Yang1094 (our method)	0.9428
3	Alexa	0.9336
4	LAST	0.9219
5	FSU_HLJIT	0.9157

Table 4
Comparison of different models

Model	Features	Development set accuracy
Logistic Regression	Word-1gram	0.7485
Language Model	Word-1gram	0.6312
Neural Network	Word-1gram	0.7348
Random forest	Word-1gram	0.7435
Random forest	char-2+3+4+5+6+7-gram+tfidf	0.9006

Table 5
Experimental results with different feature combinations

No.	Features	Development set accuracy	No.	Features	Development set accuracy
1	Word-1gram	0.7485	14	Word-4gram+tfidf	0.8164
2	Word-2gram	0.8283	15	Word-1+2gram+tfidf	0.8084
3	Word-3gram	0.8294	16	Word-1+2+3gram+tfidf	0.8230
4	Word-4gram	0.8036	17	Word-1+2+3+4gram+tfidf	0.8398
5	Word-1+2gram	0.8206	18	Char-2gram+tfidf	0.9013
6	Word-1+2+3gram	0.8336	19	Char-3gram+tfidf	0.9029
7	Word-1+2+3+4gram	0.8162	20	Char-4gram+tfidf	0.8983
8	Char-2gram	0.8219	21	Char-2+3gram+tfidf	0.9240
9	Char-3gram	0.8224	22	Char-2+3+4gram+tfidf	0.9301
10	Char-2+3gram	0.8284	23	Char-2+3+4+5gram+tfidf	0.9327
11	Word-1gram+tfidf	0.7876	24	Char-2+3+4+5+6gram+tfidf	0.9348
12	Word-2gram+tfidf	0.8005	25	Char-2+3+4+5+6+7-gram+tfidf	0.9356
13	Word-3gram+tfidf	0.8093			

4. Conclusions

This paper introduces an n-gram-based authorship identification of source code method. This method uses the logistic regression as a multi-classifier. Through the analysis of the experimental results, it can be concluded that the model of character 2-7gram after filtering by TF-IDF method has the best result, and the final accuracy rate is 0.9428. Most source codes can identify the correct author through this method.

5. Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 61806075 and No. 61772177), and the Natural Science Foundation of Heilongjiang Province (No. F2018029).

6. References

[1] Fadel, Ali and Musleh, Husam and Tuffaha, Ibraheem and Al-Ayyoub, Mahmoud and Jararweh, Yaser and Benkhelifa, Elhadj and Rosso, Paolo. Overview of the PAN@FIRE 2020 Task on Authorship Identification of SOurce COde (AI-SOCO).Proceedings of The 12th meeting of the Forum for Information Retrieval Evaluation (FI RE 2020), 2020.