Ranking-based and Classification-based Approaches for Code Author Identification

Zhongyuan Han[a], Tang Li[b], Xiangyu Wang[b], Yujie Xu[b], Menghan Wu[b], Zhiran Li[c], Zhengyu Wu[b], Yong Han[a,*]

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

Abstract

In this paper, we propose two approaches, the ranking-based approach and the classification-based approach, for the source code author identification (AI-SOCO) task of FIRE2020 (Forum for Information Retrieval Evaluation). The ranking-based approach ranks the source codes according to the number of occurrences of the 15-grams, while the classification-based approach exploits the TF-IDF of terms as features to learn a classifier. Although the ranking-based approach is very simple, it gets an accuracy of 0.9157 in the evaluation.

Keywords

Ranking, Classification, Code Author Identification

1. Introduction

The ability to identify the original author of a certain source code is needed in various domains, such as open-source projects on public platforms, mainstream online programming contest, programming examination, academic plagiarism, and shift responsibility from a fatal coding bug. In 2020, FIRE released a code author identification task, named Author Identification of SOurce COde (AI-SOCO): AI-SOCO aims to identify the author of a given source.

Code author identification task is usually regarded as a multi-classification task. For AI-SOCO task, we considered not only the multi-classification model but also the ranking model. Although the ranking-based approach has not yet achieved the state-of-the-art performance, we think it is still a valuable research directions.

2. The proposed approaches

Similar to the existing methods, our the first approach is based on the classification-based approach. We extracts the TF-IDF feature of each source code as the input and the different authors as labels to train a multiclass classification model. We choose the Random Forest[1,2] as the learning algorithm to learn the classifier. In prediction sessions, inputting the TF-IDF feature of a source, we predict its author according to the classification result.

Another approach is a ranking-based approach. The difference between ranking-based approach and the classification-based approach is that our ranking-based approach regards the AI-SOCO task as a ranking problem. The motivation using ranking-based approach stems from our research on plagiarism detection and microblog filtering. In these tasks, we found that ranking-based model was more effective than classification-based model[3-6]. We merge the codes written by the same author, denoted as d_i, where i represents the codes of i-th author. Firstly, we compute the similarity of a given source code (denoted as q) with the codes d_i of different author i. Then we rank each d_i according to

Forum for Information Retrieval Evaluation 2020, December 16–20, 2020, Hyderabad, India
EMAIL: hanyong@fosu.edu.cn (*corresponding author)
ORCID: 0000-0001-8960-9872 (A. 1)

© 2020 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

CEUR-WS.org/Vol-2826/T5-6.pdf

the similarity score. Lastly, the author of the top document is chosen as the result. For the similarity computation, we tried the number of character-based n-gram occurrences.

3. Experiments

3.1. Dataset

AI-SOCO provides the dataset consisted of 100,000 codeforces source codes (from 1,000 different authors, 100 sources per author). These codes are correct, bugless and coded in C++. 50,000 source codes(train dataset) are allowed to be used to train models. 25,000 source codes(validation dataset) can be only used to select models. And the rest 25,000 source codes(test dataset) are used to test.

3.2. Evaluation metrics

The task is evaluated by using Accuracy.

3.3. Model selection

3.3.1 Model selection of classification-based approach

We use the TfidfVectorize tool provided by scikit-learn to convert text data into TF-IDF feature vectors, and use the classifiers such as OneVsRestClassifier(LinearSVC), DecisionTreeClassifier, LogisticsRegression, KNeighborsClassifier, RandomForestClassifier (oob_score=False, random_state =None) to train our approaches with default parameters. The results are in Table 1. According to the tabel 1, we choose the Random Forest as the classifier.

Table 1.
The performance of the classification-based approach on validation dataset

Model	Accuracy
OneVsRestClassifier	0.74392
DecisionTreeClassifier	0.69263
LogisticsRegression	0.65084
KNeighborsClassifier	0.45416
RandomForestClassifier	0.80108

3.3.2 Model selection of ranking-based approach

In the ranking-based approach, we ranked each d according to the similarity score. For the similarity measure, we tried two methods. One was to use the traditional vector space retrieval model. In this approach, the vector space model was applied to compute the similarity. In our experiments, the vector space model was calculated by using Lucene (an information retrieval toolkit) with the default parameters. Another apporach was to rank the authors according to the number of occurrences of character-based n-gram in the codes of authors and the given forecasted code.

Table 2 and Table 3 show the performances of the ranking-based approaches with different number of n-gram occurrences and vector space model respectively. From table 2 and table 3, we can see that the performance of the approach using vector space model is significantly lower than that of the approach using the number of n-gram occurrences. We also note that the approach based on vector space model achieves the poor performance when using terms as features. But when introducing n-gram as features, the performance gets the greater improvement. As showed in table 3, the performance of 4-grams, 5-grams and 6-grams is lower than n-grams(n>10). It maybe that the n-grams has no ability to catch the long distance character features when n is set as a smaller value. To some extent, it shows that the n-grams(n is set as small value) can not express the writer's coding style well. Finally, we submitted the results based on 15-grams occurrence and 20-grams occurrence.

Table2.
The performance of the ranking-based approach with n-gram occurrences

Model	Accuracy
4-grams	0.8274
5-grams	0.86624
6-grams	0.88588
10-grams	0.91332
15-grams	0.91964
20-grams	0.91576
25-grams	0.90824

Table 3.
The performance of the ranking-based approach with vector space model

Description	Accuracy
Include numbers when indexing with word	0.60524
Without numbers when indexing with word	0.60484
Include numbers when indexing and remove special characters with word	0.6165
Include numbers when indexing with 2-,3-,4-,5-grams	0.6862
Include numbers when indexing with 2-,3-,4-,5-,6-grams	0.69732
Include numbers when indexing with 2-,3-,4-,5-,6-,7-,8-grams combing all the author's code	0.75564
Include numbers when indexing with 2-,3-,4-,5-,6-,7-,8-grams combing all the author's code and remove special characters.	0.77124
Include numbers when indexing with 2-,3-,4-,5-,6-,7-,8-,9-grams combing all the author's code and remove special characters.	0.77964

3.4. The performance of our submitted results

We submit three groups of results. The experiment results of our submitted results on test data are shown in the following Table 4. And Table 5 shows the best result of top 5 team[3].

Table 4.
The performance of our submitted results

Model	Accuracy
15-grams	0.9157
20-grams	0.9105
RandomForest	0.8025

Table 5.
The performance of top 5 team

#	Team	Accuracy
1	UoB	0.9511
2	yang1094	0.9428
3	Alexa	0.9336
*	AI-SOCO RoBERTa Code Baseline (6L, 12H)	0.9288
4	LAST	0.9219
5	FSU_HLJIT	0.9157

4. Discussion and Conclusions

The ranking-based approach and the classification-based approach are proposed for AI-SOCO task of FIRE2020. The ranking-based approach ranks the source codes according to the number of occurrences of the character-based n-gram, while the classification-based approach exploits the TF-IDF of terms as features to learn a classifier. Although the ranking-based approach is very simple, but it gets an acceptable performance. It shows that longer n-grams can catch the author's coding profile effectively. However, the problem of using only n-gram as features is that the occurrence of n-gram is too single to express the code profile. For example, not only the the length of code but also some global factors are not considered in n-gram based approach. Which granularities are more suitable for code author identification remains further research.

In addition, for the ranking-based approach proposed in this paper, so many things have been left unfinished, because of the lack of time. The experiments are inefficient and the performance do not meet our expectation. Some approaches based generative model or discriminative model has not been attempted in the evaluation. In future, we will plan to further develop the ranking-based model to improve the performance of AI-SOCO. For instance, using the language model model the authors' profile or using learning to rank algorithm to learn a ranking model to rank the authors.

5. Acknowledgements

This work is supported by National Social Science Fund of China (No.18BYY125).

6. References

[1] L. Breiman, "Random forests", Machine Learning, 45(1), 5-32, 2001

[2] P.Geurts, D. Ernst., and L. Wehenkel, "Extremely randomized trees", Machine Learning, 63(1), 3-42, 2006

[3] Lei-lei KONG, Zhong-yuan HAN, Hao-liang QI, Mu-yun YANG. Source Retrieval Model Focused on Aggregation for Plagiarism Detection. Information Science. 2019,503: 336–350.

[4] Leilei Kong, Zhongyuan Han, Haoliang Qi, and Zhimao Lu. A Ranking-based Text Matching Approach for Plagiarism Detection. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences. 2018, 101: 799-810.

[5] Leilei Kong, Zhimao Lu, Zhongyuan Han, Haoliang Qi. A ranking approach to source retrieval of plagiarism detection. IEICE Trans. Information and Systems. 2017, E100-D(1):203-205.

[6] Han Zhongyuan, Yang Muyun, Kong Leilei, Qi Haoliang, Li Sheng. A Hybrid Model to Real-time Microblog Filtering. Chinese Journal of Electronics. 2016, 25(3):432-440.

[7] 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)". In: Proceedings of The 12th meeting of the Forum for Information Retrieval Evaluation.