Design and Implementation of Automatic Exercises Evaluation System for Programming Courses

Liu Mingtao¹*, Xu Rui¹, Wang Yunyu¹, Li Zhiyang¹, Zhang Xin¹*

¹School of Information Science and Engineering, Linyi University, Linyi, Shandong, 276000, China
*liumingtao412@hotmail.com
*zhangxin@lyu.edu.cn

Abstract. The evaluation of students’ exercise process is an important part of the formative evaluation for programming courses. However, it is difficult for teachers to complete the evaluation of each student’s exercise process manually. This paper proposes a design of an automatic exercises evaluation system, which can automatically analyse the amount of students' exercise codes and help teachers to complete the evaluation. The system uses Python's Pyhook3 library to monitor the keyboard input of the student's computer. After text segmentation is performed on the student's input code, the amount of the student's exercise codes is statistically analysed, and the result can be used as an important indicator of the formative evaluation. Through the trial, it is found that there is a certain proportional relationship between the evaluation given by the system and the final test result of the students, which indicates the system can give a relatively objective evaluation of the students' learning process.

1. Introduction
Programming courses have strong practical characteristics. The traditional teaching mode, which is teacher-centered, focuses on the theoretical knowledge teaching, but lacks of attention to improve students' practical ability of programming. At the same time, the traditional evaluation method which relies entirely on the final exam will lead to a worse situation where most of students only cram for the final exam at the end of the term, and pay no attention to their usual study and practice [1].

In recent years, more and more programming courses are taught in laboratory instead of traditional classroom. In order to train students' hands-on coding ability, a large number of practical exercises are often interspersed in the teaching process in the laboratory. Students' performance in these practice sessions partly reflects their learning level of the course. Therefore, if these practice sessions can be considered as a part of the formative evaluation of the course, students can be urged to participate in the practical exercises more actively, and the overall evaluation system of the course can be improved to better reflect the learning process of students [2]. At present, the biggest difficulty in evaluating these in-class practical exercises lies in its heavy workload, especially when there are too many students in one class, which is a common situation in China. It is almost impossible for teacher to manually evaluate each student's in-class practice [3].

In view of the above problems, we believe that the evaluation should not be based entirely on the final exam, but also pay attention to the in-class practice process, which can be evaluated by real-time
collected data of students’ programming exercises. Based on this idea, this paper designs an automatic evaluation system of practical exercises for programming course.

2. Design of the automatic evaluation system

In order to evaluate students’ classroom practice in real time, the system needs to monitor the code information inputted by keyboard, so C/S architecture is adopted. The server-side is installed on the teacher’s computer and is responsible for setting up the system, collecting and analysing the evaluation results. The client-side is installed in the students’ computers, which will monitor the students’ keyboard input in real time, and upload students’ programming codes to the server-side. The overall architecture is shown in Figure 1.

![Figure 1 The overall architecture of the system](image)

The server-side program is functionally divided into three functional modules, as shown in Figure 2. The setup module is responsible for the initial setting of the system, such as the network parameters, course and student information, program language keywords, etc. The monitoring module is responsible for receiving and storage of information uploaded by multiple clients. The multithread programming technique is used to fulfil the real-time requirements. The score processing module is responsible for displaying, exporting and visually analysing the students’ code information collected before.

![Figure 2 Program structure of the server-side](image)

The client-side program will be deployed on students’ computers, which realizes the monitoring and data transmission of students’ keyboard input. When the client-side program starts to run, it tries to connect to the server-side firstly, and then students can log in to the system by name and student Id. After successful login, the program is minimized to the background to prevent from closing unexpectedly. During the running process of the program, the keyboard hook technology is used to
record the information input by students, and the text analysis is carried out based on keywords to obtain the code amount, and the analysis results are uploaded to the server at a certain time interval until the final shutdown. The specific process is shown in Figure 3.

![Figure 3 Flow chart of the client-side](image)

**3. Implementation of the automatic evaluation system**

3.1. Selection of development technology

This system is developed using Python language. Python is one of the most widely used programming languages in IT industry at present. In particular, there are a large number of open source third-party libraries in Python language community that can be used directly, which can greatly simplify the system development process. According to the requirements of this system, we choose Pyhook3 library to realize keyboard hook, tkinter library to realize form design, socket library to realize network communication based on C/S architecture, and openpyxl library to realize data import and export. For data storage, SQLite is used as the database. SQLite is a lightweight database, which can be easily accessed by Python language to achieve data persistence.

3.2. Implementation of keyboard monitoring feature

Keyboard monitoring is one of the core features of this system. On the client-side of the system, this function is responsible for capturing the information entered by the keyboard in the process of student programming, and sending the monitoring information to the server-side in real time for further analysis and statistics. This system uses PyHook3 library to realize the keyboard monitoring feature in the client-side. PyHook3 library provides callback hook function entry for global mouse and keyboard events in Windows [4]. With PyHook3, an event handler can be registered for some user input events (such as keyboard pressing). When the event happens, the callback hook function automatically executes to realize the monitoring of keyboard input. The programming flow is shown in Figure 4.

![Figure 4 Flow chart of keyboard monitor program](image)

It should be noted that Pyhook3 is based on Window system API, so pywin32 library needs to be installed in advance, otherwise the keyboard hook call will raise an error.
3.3. Program keyword analysis based on text segmentation

The program keyword analysis based on text segmentation is also one of the innovations of this work. The client program captures the information entered by the students and transmits it to the server-side. The server will use text segmentation technology to analyse the number of keywords to objectively estimate the grade of each student.

This system uses the Jieba library to realize the text segmentation of the input information. Jieba library is one of the best text segmentation components in Python, which supports three text segmentation modes: exact mode, full mode and search engine mode, and also supports a self-defined text segmentation dictionary [5]. In order to achieve a higher accuracy of text segmentation for program keywords, all keywords of Python are added to the self-defined text segmentation dictionary. It is easy to get the number of program keywords by the segmentation, which are accumulated over student Id and stored in SQLite database. Finally, the number of program keywords entered by all students can be obtained, which can be used as a criterion for estimating the grade of each student's exercises. The process is shown in Figure 5.

4. Analysis on implementation effect of the system

The system has been used in the “Python Language Programming” Course in the second semester 2018-2019 in School of Information Science and Engineering of Linyi University. The relationship between students' score in final exam and the evaluation grade given by the system is shown in Figure 6. As can be seen from the figure, there is an obvious positive relationship between them, and most students who write more code in the usual practical exercises have better final scores. It indicates the system can give a relatively objective evaluation of the students' learning process.

On the other hand, the abnormal values can also provide guidance for our personalized teaching. For example, the No.26 and No.27 students in Figure 6 have achieved good results in final exam, but the evaluation grade by the system is very low, which means more attention should be paid to these two students to find out the reasons behind abnormal values.
5. Conclusion
In this paper, an automatic exercises evaluation System for programming courses is designed and implemented. Through the trial in the Python course in Linyi University, it is found that the evaluation grade of the system is proportional to the final test results of students, which shows the automatic evaluation of coding practice by the system can objectively reflect students' learning process. With the help of this system, the formative evaluation can effectively motivate students to pay more attention to the program practice in usual, which will improve the learning effect and students’ practical ability of coding. But there are still some shortages of this system. For example, the system can be only used in laboratory class, and it is impossible to collect data of students' self-learning process after class. How to obtain the information of students' self-study process after class so as to evaluate students' learning process more objectively will be one of the research directions in the future.

Acknowledgments
This article is one of the phased achievements of 2019 College Student Innovation Training Project of Linyi University "Automatic Exercises Evaluation System for Programming Courses". At the same time, it is also supported by the Shandong Province First-Class Undergraduate Courses Project "Foundations for Programming Languages" (51320060) and Linyi University Doctor Start-up Fund Project "The Study of Ontology-Based Personalized Exercise Recommendation System" (LYDX2016BS114).

References
[1] Liu M, Wang X, Ding L, Zhang L. (2019). Construction and Application of Formative Assessment System of Software Engineering. Proceedings of the 2019 4th International Conference on Modern Management, Education Technology and Social Science (MMETSS 2019).
[2] Zeng X, Jiang X, Wang D, et al. (2019). Design and practice of course evaluation system based on engineering education accreditation——taking "polymer physics" as an example. Polymer Bulletin.
[3] Xu X, Li G, Guo X. (2019). Thinking and construction of process learning evaluation system from the perspective of effective learning [J]. Education Teaching Forum, 000(001), 218-220.
[4] Wang X, Zhang Y, Li S. (2014). Keyboard and mouse monitoring system based on WIN32. Software Guide, 000(001), 99-100,101.
[5] Yan M, Zheng C. (2018). Text segmentation and word cloud production in python environment. Modern Computer, 000(034), 86-89.