IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

A Literature Survey on Automatic Evaluation of Computer Programs

Priyanka Sharma[1], Samkitkumar Jain[2], Prachin Ranawat[3], Husain Poonawala[4], Pankaja Alappanavar[5]

Student, Department of Information Technology, Sinhgad Academy of Engineering, Pune, India [1,2,3,4]

Professor, Department of Information Technology, Sinhgad Academy of Engineering, Pune, India [5]

Abstract: Today, due to a large number of online courses there is a need for automatic evaluation of computer programs. Grading of computer programs is also helpful for the companies in their recruitment process where a set of programming statements are to be solved by the students. Thus, automatic grading of open-ended responses has increasingly become a subject of research. Existing systems which are used for such assessments provide a score on the basis of a number of test cases which are passed by the computer program. This does not reflect the overall abilities of the programmer. Blending the existing systems with machine learning and considering the programming styles can prove to be more useful for the evaluation of computer programs. This paper gives a review of the literature survey done on Automatic Evaluation of computer programs.

Keywords: Program, Automatic grading, Machine Learning, Natural Language Processing.

I. INTRODUCTION

These days, in the case of company recruitment processes and also in the case of the online courses (called Massively Open Online Courses- MOOCs), students are required to solve questions related to programming. These questions are given in the form of problem statements wherein the students are required to submit their solution in the form of a code written in the programming language specified.

The solutions are then checked with the test cases specific to each problem statement and the result is generated based on the number of test cases passed. But this often leads to the rejection of some capable candidates, in the case of recruitment processes and also needs more time for grading manually, in the case of MOOCs. Hence, here arises a need to grade such computer programs according to the efficiency of the codes and not only with the test cases. It could be used to reduce the dependency on the availability of manual evaluators and also help improve self-assessment.

This paper, mentions the solutions to return the score of a candidate according to the coding efficiency along with the number of test cases passed as output as discussed in the reference papers. It has been organised into seven sections as follows. Section 2 outlines the need causing a research in this topic. Related work done in this area is presented in Section 3. The methodology of Machine Learning considering the program styles has been described in sections 4 and 5, respectively. Section 6 focuses on using web technology in this field. Conclusion and our expectations related to the future trends in automatic assessment of programming assignments are discussed in Section 7.

II. BACKGROUND

For writing good computer programs, it is necessary to understand the problem statement and accordingly follow different approaches in order to get the most effective and better solution. It is also important to consider the various styles in which the codes can be written.

There exist a number of ways in which a piece of code can be written. But, following a particular standard makes it easy for an evaluator to assess the code.Giving proper indentation, using meaningful variable names, proper use of keywords and other constructs of a programming language are few examples by which the quality of a computer program can be improved.

This makes the task of evaluation efficient and allows students to be marked on these aspects also. Hence, considering the above-mentioned benefits which can be provided to the manual evaluators and the students, there is a need of such a system that will be able to grade the students according to their level of understanding.

III. RELATED WORK

Anumber of approaches have been made to evaluate computer programs. Machine learning is one such approach which is used in Automatic Grading of Computer Programs: Machine Learning Approach by Shashank Srikant and Varun Aggarwal [1]. This machine learning approach derives a set of features from a given program to evaluate the program code. These features are then used to learn a model to grade the programs on the basis of a set of rules.

The ability to write nice and elegant programs is a very important skill which is not usually the focus of programming courses [2]. A program written in a bad style would still work and this is the reason programming style

DOI 10.17148/IJARCCE.2017.6264

is considered to be secondary by students. A programming style is considered to be an individual's interpretation of rules and their application to the writing of source code in order to achieve the aim [3]. Style can be used to ensure that the source code is readable and understandable. Automatic analysis of functional program style by Greg Michaelson describes semantic style rules and has also presented an automatic style analyser. Thus, paying more attention to the style of programming aids the evaluation of programs.

In a number of universities, often a methodology of peer-assessment is used. In this, the programs are evaluated by other students belonging to the same class. This helps in situations where the manual evaluators are not available and also allows students to understand mistakes while evaluating the codes. This method can be combined with the web technology and has been stated by Jirarat Sitthiworachart and Mike Joy [4].

A more detailed description of the work done in this field has been mentioned in the subsequent sections of this paper.

IV. MACHINE LEARNING APPROACH

Machine learning is the field in which models are constructed from the training data (input data) for decision making. It makes use of different types of algorithms that can learn from and make predictions on data. The models created are data-driven and not follow the instructions which are strictly static.

For applying machine learning framework to analyse the computer programs, different features related to a program [1] submitted by a candidate need to be extracted, which were then used to train a machine learning model.

Different types of machine learning models used are:

Supervised learning: Training of model is done by using labelled data and then tests the model for unlabeled data.

Unsupervised learning: This technique uses unlabeled data for training as well as testing.

Semi-supervised Learning: This uses both supervised and unsupervised learning models i.e.; the labelled and unlabeled datasets.

Shashank Srikant and Varun Aggarwal [1] implemented the training of the model by using two machine learning algorithms namely, Ridge Regression and Support Vector Machine. A comparison of these two methods was also done which showed that Ridge regression outperformed the other in terms of better accuracy.

Two programming problems -the Encrypt problem and the Alt Sort - were considered. A set of sample computer programs in C were taken. The sample codes were written by final year Computer Science/IT engineering students in India. Each problem had a suite of test cases for checkingbasic and advanced conditions of the logic of the problems. The rating of the programs was done on ascale of 1-5 following the mechanism is shown in the table below. This was used to assign a final grade to the submitted solution.

TABLE I SCORING MECHANISM

Score	Meaning
5	Completely correct and efficient.
4	Correct butfail to pass the code through all test-cases.
3	Inconsistent logical structures: few/partially correct data dependencies.
2	The presence of basic structures: Shows some understandingof a part of the problem.
1	Less unrelated to the problem statement.

Any program with no code or less than 5-6 lines of code which had no relevance to the problem statement was removed from the data set.

This reduced the size of the data set considerably to around 90 for both the problem statements. Due to such unrelated solutions submitted by students, the size of the dataset decreased. Also, a number of students did not attempt the questions which point to the poor ability of students in programming. After the collection of the dataset, different types of features mentioned below were extracted from the code submitted.

Basic Features: These were obtained by counting the occurrences of various keywords and tokens appearing in the source code. These include keywords related to control structures such as for, while, break, etc.

Expression Features: includes features obtained by counting the occurrences of expressions appearing in a program. The expressions used in a program help identify arithmetic and relational operations typical of the underlying algorithm.

Basic & Expression Features in Control Context: This includes identifying the above features in the context (block scope) of the control-flow structures (conditional statements or loops).

Data-Dependency Features: This class includes features obtained by counting the occurrence of particular kind of expressions which are dependent on other particular kinds of expression. Such an ordering between two expressions generally signifies that the value of a variable(s) in one expression influences the evaluation of the other expression.

V. PROGRAM STYLE

G Michaelson, Automatic Analysis of functional program style [2], mentions that program style is concerned with how effectively an input/output process is expressed, regardless of what that process actually is. A poor style is attributed to either writing too much with the unnecessary details or writing too little hiding the necessary details. Considering the variations in style, programming languages have a lesser scope of variations as compared to

DOI 10.17148/IJARCCE.2017.6264

the natural languages. A number of people have contributed in providing the styling guidelines, for example, guidelines for Pascal have been given by Marca [5].

In industries, as a project advances in the further stages of development, it becomes harder to maintain it if different programming styles have been used by those involved in its maintenance. Thus requiring, following a standard style or using a style which is consistent throughout the code [3]. Thus, students should be made to write their programs in a manner which can be comprehended easily starting from their school level itself.

The errors in the style of writing codes are an indicative of misunderstandings in the concepts by novices. Therefore, checking for the ways in which a piece of code is written helps to grade the program in an efficient manner.

Testing the typographical style of a set of COBOL programs was done by Mohan and Gold, 2004 [3]. The analysis was done by making comparisons to the results found by a manual checker with those done by a tool. The results showed that the typographical changes in style were proportional to the maintenance of programs done across various versions.

Some rules have also been devised to check the programming styles. Rules to simplify the Boolean comparisons with conditional statements are mentioned along with the removal of the unused variables in computer programs [2]. The removal is done by replacing the variable with a certain character which is set by default in the style checker system for the analysis of computer programs. The system is capable of providing some style suggestions as well.

VI. WEB-BASED ASSESSMENT

Grading of computer programs by other students is often used at schools and universities to encourage independent learning amongst students. It is possible to do this peer assessment using the web technology.

Sitthi Worachart, J., and Joy, M., in Web-based Peer Assessment with an Anonymous Communication Tool [4], describe such a system which uses a communication tool to allow students assess the computer programs online. The system allows them to send messages, which can be viewed offline as well; in case someone is not online. The identity of the students marking the programs is kept anonymous.

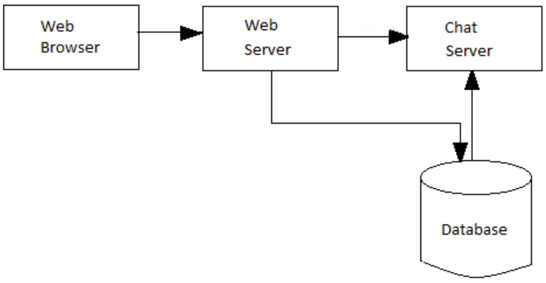

Fig. 1. System Architecture

The system is monitored by an administrator to ensure the efficiency of the tool and keep a check on specific interactions like asking for the student identity or use of fowl words. This monitoring can be done only by the system administrator to ensure privacy.

From the architecture, it can be seen that through the web browser, student responses are taken and are forwarded to the server which is later stored in the database. The database is used by the chat server and the web server for evaluation.

The communication device displays the programs which are needed to assess the students along with two anonymous students who would be evaluating those same programs.

Such systems provide an alternative way for the evaluation of computer programs.

VII. CONCLUSION

This literature survey was done with the purpose to study and understand the different methodologies that can be used for automatic evaluation of computer programs. The paper discusses, the various aspects of building an evaluation system, namely using the programming style to construct a Machine Learning model for assessment.

Paper represented by Shashank Srikant and Varun Aggarwal [1] specifies the use of a machine learning framework with the help of different features for evaluating computer programs automatically.

The style in which a program has been written is also beneficial in order to grade the programs as mentioned in Automatic Analysis of functional program style by G.Michaelson [2].

Sitthi Worachart, J., and Joy, M., Web-based Peer Assessment System with an Anonymous Communication Tool[4] has put forth a system which allows students to grade their programs and also provides them with the ability to discussany queries with the other students.

Further, we are continuing this research by using other Machine Learning algorithms such as the Random Forest Algorithm. We would be focusing on C and C++ programs, albeit a system working for the other programming languages can also be constructed.

ACKNOWLEDGMENT

Wetake this opportunity to thank all the people involved in the making of this paper. We want to especially thank our respected guide, Ms Pankaja Alappanavar for her guidance and encouragement, which has assisted us to achieve our goal. Her valuable advice has helped us throughout. Our Head of Department Prof. Abhay Adapanawar has also been very helpful and we are grateful for the support he provided us with. Last but not the least we would like to convey our gratitude to all the teaching and non-teaching staff members of our department, our friends and families for their valuable suggestions and support.

DOI 10.17148/IJARCCE.2017.6264

REFERENCES

[1] Shashank Srikant and Varun Aggarwal, Automatic Grading of Computer Programs: A Machine Learning Approach, 12th International Conference on Machine Learning and Applications (2013).

[2] G. Michaelson, Automatic Analysis of functional program style, Software

[3] A. Mohan and N. Gold. Programming style changes in evolving source code.2004

[4] Sitthi Worachart, J., and Joy, M., Web-based Peer Assessment System with an Anonymous Communication Tool, Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT'04).

[5] D.Marca. Some Pascal style guidelines. SIGPLAN,16(4):70-80, April 1981. Metevand V. P. Veiko, Laser-Assisted Microtechnology, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.

BIOGRAPHIES

Priyanka Sharma Pursuing Bachelor of Engineering (B.E) in Information Technology from Sinhgad Academy of Engineering, Savitribai Phule Pune University (S.P.P.U)

Samkitkumar Jain Pursuing Bachelor of Engineering (B.E) in Information Technology from Sinhgad Academy of Engineering, Savitribai Phule Pune University (S.P.P.U)

Prachin Ranawat Pursuing Bachelor of Engineering (B.E) in Information Technology from Sinhgad Academy of Engineering, Savitribai Phule Pune University (S.P.P.U)

Husain Poonawala Pursuing Bachelor of Engineering (B.E) in Information Technology from Sinhgad Academy of Engineering, Savitribai Phule Pune University (S.P.P.U)

Pankaja Alappanavar ME Computer Engineering, Sinhgad Academy of Engineering, Pune

DOI 10.17148/IJARCCE.2017.6264