International Journal of Digital Crime and Forensics
Volume 15 • Issue 1

The Metric for Automatic Code Generation Based on Dynamic Abstract Syntax Tree

Wenjun Yao, Kunming University of Science and Technology, China
Ying Jiang, Kunming University of Science and Technology, China*
Yang Yang, Kunming University of Science and Technology, China

ABSTRACT

In order to improve the efficiency and quality of software development, automatic code generation technology is the current focus. The quality of the code generated by the automatic code generation technology is also an important issue. However, existing metrics for code automatic generation ignore that the programming process is a continuous dynamic changeable process. So the metric is a dynamic process. This article proposes a metric method based on dynamic abstract syntax tree (DAST). More specifically, the method first builds a DAST through the interaction in behavior information between the automatic code generation tool and programmer. Then the measurement contents are extracted on the DAST. Finally, the metric is completed with contents extracted. The experiment results show that the method can effectively realize the metrics of automatic code generation. Compared with the MAST method, the method in this article can improve the convergence speed by 80% when training the model, and can shorten the time-consuming by an average of 46% when doing the metric prediction.

KEYWORDS

Automatic Code Generation, Dynamic Abstract Syntax Tree, Extract Algorithm, Metric

INTRODUCTION

Automatic code generation technology is one of the means to improve the efficiency and quality of software development. Many researchers have studied technical means and improved automatic code generation implementation methods to enhance the quality of automatically generated code and achieve the purpose of meeting programmer's expectations. Therefore, automatic code generation technology has always been the core issue for practitioners and researchers (Hu et al., 2019).

In recent years, artificial intelligence technology has made great progress and development, which has further formed an important promotion for the research on automatic code generation technology. Automatic code generation technology has also achieved vigorous development (Yang et al., 2020). Part of the current research on automatic code generation technology has been applied to actual development. The automatic code generation tools implemented according to a certain code

DOI: 10.4018/IJDCF.325062

*Corresponding Author

generation method are usually embedded in the integrated development environment in the form of plugins. For example, integrated development environments such as IntelliJ IDEA, Eclipse, and PyCharm all support embedded code automatic generation plugins to help programmers improve development efficiency.

Among the number of automatic code generation technologies, the ability to evaluate the model includes the accuracy of predicting the next token, the MRR indicator used in the field of information retrieval, and so on. The evaluation indicators cannot be directly converted because of automated evaluation standards are different (Hu et al., 2019). Therefore, the existing research on automatic code generation technology focuses on the improvement of its model capability, ignoring the research on the quality measurement of its generated code. In addition, the existing research on software code quality measurement is based on the content of the written code, ignoring the programmer's programming behavior information. However, programming behavior information is a factor that cannot be ignored when doing the automatic code generation measurement. The programming behavior information is continuously and dynamically changing. So the automatic generation of code measurement is also a process of continuous dynamic changeable. Traditional software quality measurement methods are not suitable for code measurement generated by code automatic generation technology. Therefore, it is of great significance to study the metric method for the code generated by the automatic code generation technology.

To fully consider that the automatic code generation metric is a process of continuous dynamic changeable, this paper proposes a metric method for automatic code generation based on dynamic abstract syntax tree (DAST method). Specifically, the authors build a dynamic abstract syntax tree, and then extract metric-related content from the dynamic abstract syntax tree. Finally, the authors complete the metric for automatic code generation according to the extracted content. The experiment results show that the method can effectively realize the metrics of automatic code generation. Compared with the automatic generation metric method (Zhang, 2021) (MAST method) of which constructed by all programming codes and programming records, the method in this paper can improve the convergence speed by 80% when training the model, and can shorten the time-consuming by an average of 46% when doing the automatic code generation metric prediction.

The contributions of this paper are summarized as follows:

1. In this paper, the authors propose an algorithm to construct a Dynamic Abstract Syntax Tree (DAST) by combining programmer behavior and code generation tool behavior information. This algorithm effectively utilizes the cooperation behavior information between programmers and code generation tools in the programming process. A DAST reflects the dynamic changes of semantics, structure information and programming behavior information in a abstract syntax tree (AST), providing a basic platform for the measurement of code generated by code automatic generation tools.
2. The authors propose an algorithm to extract metric content through traversing the DAST. This algorithm effectively extracts the content which is deeply related with metric for automatic code generation.
3. The authors propose a method to build a metric model for automatic code generation based on code semantic, programmer behavior and behavior information of code automatic generation tools, and send the extracted content to this model and complete the measurement.

The rest of this article is organized as follows. First this article introduces related work on automatic code generation techniques and metric for automatic code generation. Secondly this article elaborates the construction and implementation of DAST. Thirdly this article elaborates algorithms for extracting metric content on DAST. Then this article elaborates the method of automatically generating metrics from code based on the extracted metrics content of DAST. After that the article shows the experimental setup and analyzes the results. Finally, this article concludes this article's work.