Automated Error Detection Through Stack Overflow With The Help Of Python Tool

K.David Raju (Associate Professor)

Department of Computer Science and Engineering

St. Peter's Engineering College

Hyderabad, India

Akhil.L (UG Scholar)

Department of Computer Science and Engineering

St. Peter's Engineering College

Hyderabad, India

G. Shruthi (UG Scholar)

Department of Computer Science and Engineering

St. Peter's Engineering College

Hyderabad, India

V.Venkat (UG Scholar)

Department of Computer Science and Engineering

St. Peter's Engineering College

Hyderabad, India

Abstract—As a popular Q&A site for programming, Stack Overflow is a treasure for developers. However, the number of questions and answers on Stack Overflow make it difficult for developers to efficiently locate the information they are looking for. There are two gaps leading to poor search results: the gap between the user's intention and the textual query, and the semantic gap between the query and post content. Therefore, developers have to constantly reformulate their queries by correcting misspelled words, adding limitations to certain programming languages or platforms. As query reformulation is tedious for developers, especially for novices, we propose an automated software-specific query reformulation approach based on python script. Our approach trains a transformer model that can automatically generate candidate reformulated predicted relevant when given the user's original query.

Keywords—Python,Query-Resolving,Relevant-Query output,GUI,Stack API.

I. INTRODUCTION

Debugging is a tedious task for the beginners, Debugging can help programmers or developers pinpoint various types of errors, semantic errors, syntax errors, logical errors. Often times testing and debugging software takes a longer time than writing the software itself. To prevent incorrect operation of a system or software, debugging is used to find and resolve bugs or errors. Debugging tactics can involve interactive debugging is the process of finding and resolving errors within the computer programs, software or system. Debugging tactics can involve interactive debugging, control flow analysis, unit testing, integration testing, log file analysis, monitoring at the application or system level, memory dumps, and profiling. Many programming languages and software development tools also offer programs to aid in debugging, known as debuggers. Debugging ranges in complexity from fixing simple errors to performing lengthy and tiresome tasks of data collection, analysis, and scheduling updates. The debugging skill of the programmer can be a major factor in the ability to debug a problem, but the difficulty of software debugging varies greatly with the complexity of the system, and also depends, to some extent, on the programming language(s) used and the available tools, such as debuggers. Debuggers are software tools which enable the programmer to monitor the execution of a program, stop it, restart it, set breakpoints, and change values in memory. The term debugger can also refer to the person

who is doing the debugging. In certain situations, general purpose software tools that are language specific in nature can be very useful. These take the form of static code analysis tools. These tools look for a very specific set of known problems, some common and some rare, within the source code, concentrating more on the semantics (e.g. data flow) rather than the syntax, as compilers and interpreters do.

II. IMPLEMENTATION

As proposed earlier, in this prediction of solution to the errors in the program. A python script is introduced to address the errors occurred in the program code to provide a relevant answer answered on Stack API which pops a browser and gives the results. We have used Python, Stack Overflow.

A. Python

Python is our programming language. It is used for creating a Graphical user Interface (GUI). It provides us an interface to access our script and file. It provides various tools and libraries that help consistently creating our model.

B. Stackoverflow

Stack overflow is Q&A website for professional and enthusiast programmers. Here using our python script, we fetch relevant solutions for the errors occurred in the program code.

III. WORKING MODEL

A. Input Collection and Preprocessing

We take input from the user as a code file and train our model to detect the error in the code a fetch the reformulates relevant solutions through stack overflow platform. Here after collecting input our python script detects the error and if the error found on stack overflow it pops up a default browser ad provides relevant solutions to the user.

- It provides a relevant solution to the user so it gives

- The website serves as a platform for users to ask and answer questions, and, through membership and active participation, to vote questions and answers up or down similar to Reddit and edit questions and answers in a fashion similar to a wiki. Users of stack overflow can earn reputation and badges

- It provides a relevant solution to the user so it gives a basic overview of the error and user gains the knowledge on the error as WHAT, WHY, HOW these kind of analysis to the user

- Workflow
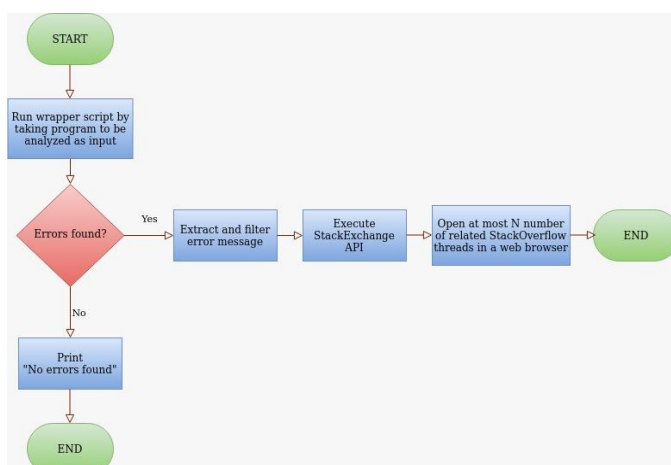
B. Building up a Python Script

This proposed model is first prepared to accept user input code file it will accessed from the GUI, when the code file is accepted by the system architecture of our model. The model checks the whole code completely and compiles and interpreted and give the compilation results so that if an error occurs it would be going in a different path if there is no error it would be going in different one. From this python script we will be detecting is there are any errors and for accessing the code file the methodology interlinked is here a GUI is implanted so there is an interface to access the file and run the file to get proper output. To develop a GUI, we used Tkinter here and some more libraries are used to build these Python Script. Because of this script we have built a tool which automatically detects the error occurred in a python program so that the user will be knowing what kind of error it is so that he would be getting a solution to his error to resolve his problem which will be a good initiative to learn his course or program. By this script the resulted following progressive output which shows the type of error like syntax, name error, etc.


C. Bridging the gap between script- Stack Overflow

- In this after the working of the script we just need a reformulated relevant solution to the error found in the code file given by the user, to bridge this gap we use stack overflow platform it is a question & answer platform where enthusiast programmers ask and send the queries and solutions to the peer ones. For that we should built a bridge for our script and stack overflow to get our query solved. For that we extract the Stack overflow API so that we add the bridge to get our query to solved so that it would be a good one to solve the problem. Building a bridge using this API that our script will fetch the reformulated relevant solutions to the query solved. From this project the users will be getting solved their queries and gaining some sort of knowledge regarding their query. So that the bridging the gap would be done here to fetch the relevant solutions easily.

D. Application

The application starts with python and goes with a interface gui and here now actual process go on as interpretation and compilation of python code occurs and sends back the errors to stack overflow and fetch some reformulated and relevant solutions to the user via default browser in their laptop/desktop.

E. RESULT
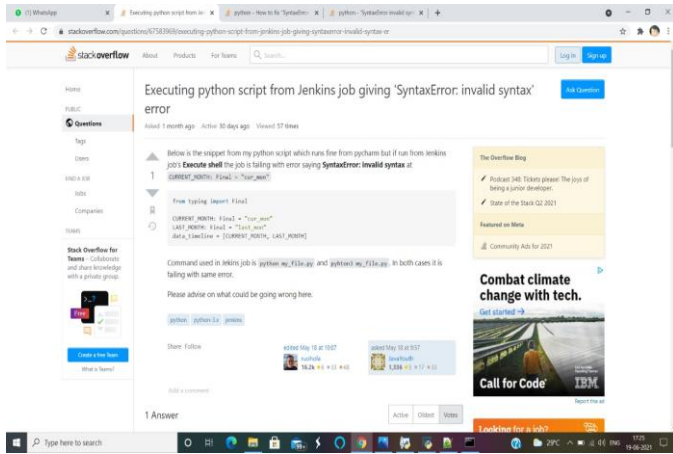
CONCLUSION

The predominant motive of this research is to provide an Error Detection Using Stack Overflow and providing a reformulated relevant mostly highly rated or voted and highly viewed by the programmers to the users. The current approach has some things to be changed and to provide a error to their solutions to the user. We have proposed and developed a model that gives a reformulated relevant solution actively rated and viewed one to the users.

FUTURE ENHANCEMENT:

There is a degree for further upgrades and enhancements. Progressively effective approaches to incorporate different highlights and functionalities should, for that here Deep Learning, Machine learning would be great enhancement which would be a great change in relevant solutions to effective and exact solution to the error this will be a future model of our project.

References

[1] .Y. Huang, C. Chen, Z. Xing, T. Lin and Y. Liu, "Tell them apart: distilling technology differences from crowd-scale comparison discussions", 2018 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE), pp. 214-224, 2018.

Show in Context View Article Full Text: PDF (576KB) Google Scholar)

[2] C. Chen and Z. Xing, "Mining technology landscape from stack overflow", Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, pp. 1-10, 2016.

Show in Context Google Scholar.

[3] C. Chen, Z. Xing and L. Han, "Techland: Assisting technology landscape inquiries with insights from stack overflow", 2016 IEEE International Conference on Software Maintenance and Evolution (ICSME), pp. 356-366, 2016.

Show in Context View Article Full Text: PDF (582KB) Google Scholar.

[4] R. Abdalkareem, E. Shihab and J. Rilling, "What do developers use the crowd for? a study using stack overflow", IEEE Software, vol. 34, no. 2, pp. 53-60, 2017.

Show in Context View Article Full Text: PDF (728KB) Google Scholar.

[5] .X. Xia, L. Bao, D. Lo, P. S. Kochhar, A. E. Hassan and Z. Xing, "What do developers search for on the web?", Empirical Software Engineering, vol. 22, no. 6, pp. 3149-3185, 2017.

[6] C. Chen and Z. Xing, "Towards correlating search on google and asking on stack overflow", Proceedings of 2016 IEEE 40th Annual Computer Software and Applications Conference, pp. 83-92, 2016.

Show in Context View Article Full Text: PDF (698KB) Google Scholar.

[7] 9.C. Chen, Z. Xing and X. Wang, "Unsupervised software-specific morphological forms inference from informal discussions", 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE), pp. 450-461, 2017.

Show in Context View Article Full Text: PDF (488KB) Google Scholar

[8] 10.X. Chen, C. Chen, D. Zhang and Z. Xing, "Sethesaurus: Wordnet in software engineering", IEEE Transactions on Software Engineering, 2019.

Show in Context Google Scholar

[9] 11.B. J. Jansen, D. L. Booth and A. Spink, "Patterns of query reformulation during web searching", Journal of the american society for information science and technology, vol. 60, no. 7, pp. 1358-1371, 2009.

Show in Context CrossRef Google Scholar

[10] 12.M. Sloan, H. Yang and J. Wang, "A term-based methodology for query reformulation understanding", Information Retrieval Journal, vol. 18, no. 2, pp. 145-165, 2015.

[10] 13.L. Bing, W. Lam, T.-L. Wong and S. Jameel, "Web query reformulation via joint modeling of latent topic dependency and term context", ACM Transactions on Information Systems (TOIS), vol. 33, no. 2, pp. 1-38, 2015.

Show in Context Google Scholar

[11] 13.L. Bing, W. Lam, T.-L. Wong and S. Jameel, "Web query reformulation via joint modeling of latent topic dependency and term context", ACM Transactions on Information Systems (TOIS), vol. 33, no. 2, pp. 1-38, 2015.

Show in Context Google Scholar

[12] [12] 14.J.-Y. Jiang, Y.-Y. Ke, P.-Y. Chien and P.-J. Cheng, "Learning user reformulation behavior for query auto-completion", Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval, pp.