iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VI **Month of publication:** June 2021

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI Jun 2021- Available at www.ijraset.com

AI Error Resolver using CQA

Ankita Ashok Baleri[1], Nanda Krishna K S[2], Nagarjun C S[3], Suraj P[4], Nandeesh. H. D[5]
[1, 2, 3, 4]Student, Department of Computer Engineering, JSS Science and Technology University, Mysore, Karnataka, India.
[5]Asst. Prof, Department of Computer Engineering, JSS Science and Technology University, Mysore, Karnataka, India.

Abstract: ***All developers spend more amount of time fixing the errors encountered while developing a program. The AI Error Resolver (also called bugfix) is a solution to help developers fix their errors in minimal time. The objective of this solution is to develop a command line tool that dynamically suggests solutions for errors encountered in the program. The CLI tool is used to identify the error statement and the AI model selects the best answer out of multiple answers available for a question in StackOverflow platform. The AI Error Resolver mainly aims at resolving the error by providing the best solution. The AI model has a wide range of scope. It can be used by developers to resolve their errors encountered as well as the other developing team to incorporate our model in their software through a Rest API. Currently, the model is trained only for python program errors. The AI Error Resolver uses web scraping to fetch all the solutions from the website, an AI model for predicting the best solution out of many solutions and a global database for fetching the solution for recently encountered errors so that it reduces time.***

Keywords: Error Resolver, Programmer, AI, Web scraping, MLP Classifier

I. INTRODUCTION

'AI Error Resolver Using CQA' is a solution whose main motto is to help developers. In a traditional method whenever an error is encountered, developers usually choose to google for a solution. But they may end up getting multiple answers and could not be able to choose the best one. As a result, they may have to try out each and every solution until the error is resolved. This entire process takes away more time. So, our project saves this time by providing a command line tool. The developers can use it to run the program. If an error is encountered, then the error statement will be identified and sent to the global server database. A query will be executed to get the similar error recorded previously from the database. If it is found, the error along with the best solution will be returned to the user. If not, we search for the similar question in StackOverflow and collect all the answers for that question. These answers will be the input for our trained AI model. The AI model finds the best solution among the multiple answers available and inserts it to the database. Then, the error along with the best solution is written on to the console. The whole process occurs in a fraction of second. Hence, developers need not waste their time in searching for the solution.

II. LITERATURE REVIEW

There have been several previous works done to find the best answers from CQA sites. There are several approaches that were followed by different researchers like Random Forest, Feed forward NN, Logistic Regression, Support Vector Machine, Receiver Operating Characteristics Area Under Curve, Precision Recall Area Under Curve etc. Some of the previous works are given below. A fair amount of research has been done on different aspects of this project domain.

1) In 2016 Maximilian Jenders, Ralf Kresteland Naumann introduced the model to predict the best answer from the Massive Open Online Courses (MOOC) Forums. This method uses information such as the number of previous answers, number of accepted answers posted by the answerer, sum of votes received by an answerer, and number of posted comments by the answerer. Random forest and Feedforward NN are the algorithms used here. 408 answers were correctly classified as accepted answers and 11 were misclassified in Random Forest. Predicting the best answer as soon as one posts an answer without delay was challenging and referred to future enhancement.

2) Ziyu Yao, Daniel S. Weld, Wei-Peng Chen, and Huan Sun developed the model to systematically mine question-code datasets from stack overflow. It uses Bi-View Hierarchical Neural Network which considers both the natural language written along with the code snippet and the programming language available in the Stack Overflow to predict which is the standalone code solution for the given question using Logistic Regression. This paper currently only considers a code snippet to be a standalone solution or not. In many cases, code snippets in an answer post serve as multiple steps and should be merged to form a complete solution.

©IJRASET: All Rights are Reserved

3) Chethana V and EvlinVidyuLatha P have developed a similar model to predict the accepted answers of questions in Community Question Answering (CQA) sites using Receiver Operating Characteristics Area Under Curve, Precision Recall Area Under Curve and F-score. Prior involvement of the answerer on question tags and topics increases the chance to give the answer for that question. Expertise will increase the chance in acceptance of the answer.

4) Ahmad Diyanti, Behrooz Shahi, Syed Mastafa, Mohammed Hadi, Mohammed Hassan Diyanti developed similar model to predict the answers to the question in Community Question Answering (CQA) sites based on the comments on that post using K-Means Clustering. Time of posts and investigation of code can be an improvement to this model.

5) Luca Ponzanelli, Gabriele Bavota, Massimiliano Di Penta, Rocco Oliveto, Michele Lanza developed a plugin for Eclipse IDE that automatically searches the relevant discussions in the stackOverflow website and provide it to a user if the confidence threshold is surpassed.

III. THEORY

When a developer executes a program, he gets to know about the errors present in the program. Errors are the mistakes in the program that may inhibit program execution. Whenever such an error occurs, developers usually search for the solutions. To save this time, our project provides a command line tool that helps to debug an error. Command line tools are scripts, programs and libraries that have been created with a unique purpose. When the error is displayed on the interface, it not only involves the error statement but also lots of junks with it. So, basically when the error is encountered, our command line tool reads the error, removes the junk and identifies the key error statement. The database is maintained to give an instant solution for the error encountered. After the error identification stage, the query is generated to check if the same error was encountered before. If so, the best solution which was previously chosen by our model is selected from the database and sent to the programmer via command line tool. If the similar error is not found in the database, then the process is directed to the Community Question Answering platform (CQA). The CQA platform we are focussing on is 'StackOverflow'.

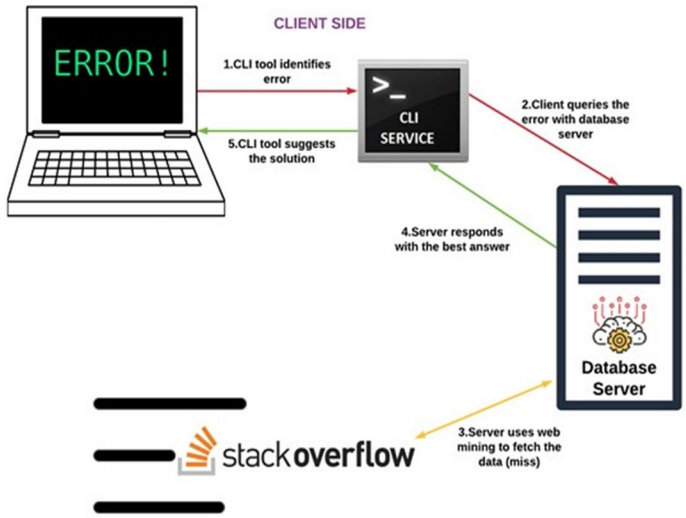


Fig.1. Flow Chart

StackOverflow is the largest and most trusted online community for developers to learn, share their programming knowledge and build their careers. It is a question-and-answer site for professional and enthusiast programmers. It features questions and answers on a wide range of topics in computer programming. The users can upvote or downvote the questions and answers. The users can also earn points on getting an upvote for the answers given by them. We are using one such platform to provide a solution for the error that a developer came across. As the process is directed to StackOverflow, the searching takes place and finds the top relevant question. Then, the question and all its respective answers are collected using a web scraping. Web scraping, web harvesting, or web data extraction is data scraping used for extracting data from websites. Scraping a web page involves fetching the data and extracting information from it and it is typically stored into a central or local database or spreadsheet, for later retrieval or analysis.

©IJRASET: All Rights are Reserved

The various features related to answer and answerer are extracted using web scraping. After getting all the parameters required, the AI model which is previously trained is used to find the best answer. The best answer chosen is stored in a database which in turn is sent to the programmer through the command line tool. The database is of fixed size to enable faster access. The developer can use the solution suggested by our model to debug an error and hence can save time.

The dataset used for training is authentic and taken from stackexchange.com website using SQL queries. The data consists of various fields like question id, question creation date, question body, title of the question, answer id, answer creation date, votes for the answer, answer body, comment count of the answers, reputation score of the answerer, overall upvotes and downvotes received by the answerer. Initially, the creation date of the question, title of the question, id of the question and answer are dropped as these fields don't add value to the model. The answer that has been accepted is considered as the best answer while output labelling. The chronological order has been considered using the creation time of the answer. The similarity among the answers has been calculated using cosine similarity. The correlation between the features is calculated and the relevant features are considered for training. The upvotes, downvotes and views are highly correlated to the reputation score. Hence the reputation score is considered and the remaining three are dropped. The attributes related to the answer have been normalised by drilling down towards each question and for other attributes overall min-max normalisation has been applied. Finally, the features considered involve user related attributes like reputation score which is also a combination of upvotes and downvotes, the content related attributes like content and code similarities of the answers and also the other attributes like number of votes, chronological order and comment counts. The normalized dataset with 8026 entries is considered for training and testing purposes. The class balance is achieved with even distribution of two classes (best, not best) and has been considered as the final pre-processed dataset. The 80% of the dataset has been used for training and remaining for testing. The AI model comes into picture when a particular question doesn’t have an accepted answer. If there is an accepted answer it will be directly returned. Similarly, if the question has only one answer, then also that particular answer will be directly returned.

The various classification algorithms are used to classify answers as best or not best answers namely Decision tree, KNN, Gaussian Naive Bayes, Random Forest, Logistic Regression and MLP Classifier. The MLP Classifier used has the 4 hidden layers and the default activation function is used for training the dataset. Among all the classification algorithms used, MLP Classifier yielded the most promising results.

IV. RESULT

The command line tool in the project is an integration of the database, web scraping and AI model. Our proposed system works as follows: Firstly, the admin maintains the database, any problem in the database will be handled by the admin. The database follows a First in First out strategy i.e. the database is of fixed size and whenever a new entry is added to the database, the oldest entry will be deleted from the database. The database has a fixed size and thus makes the searching faster. As a result, it increases the efficiency of the system and end user experience. The developer will write the code and execute it in the command line. If there is any error, the error statement along with the solution is returned. The answer is chosen by the AI model. The various classification algorithms are used namely Decision tree, KNN, Gaussian Naive Bayes, Random Forest, Logistic Regression and MLP Classifier. The accuracy obtained are as follows

Classifier	Accuracy
Decision Tree	71.73%
KNN	80.32%
Logistic Regression	82%
Random Forest	81.19%
Gaussian Naive Bayes	80.13%
MLP Classifier	82.44%

By comparing the accuracy obtained, MLP Classifier has been used to train the AI Model.

©IJRASET: All Rights are Reserved

V. FUTURE SCOPE

The bugfix is a CLI tool that reduces excessive time spent by the developer to resolve the error. It recommends a better solution for the error from the StackOverflow-CQA platform. A more advanced automation approach could be to enable bugfix to autocorrect the error in the program using the AI-recommended dynamic solutions. This prevents the burden of error resolution to the programmer, thereby bringing down the total development time.

Command Line Interface has limited User interface options. There is a scope to increase the user interaction with dynamic colours and fonts offered by the command line interface.

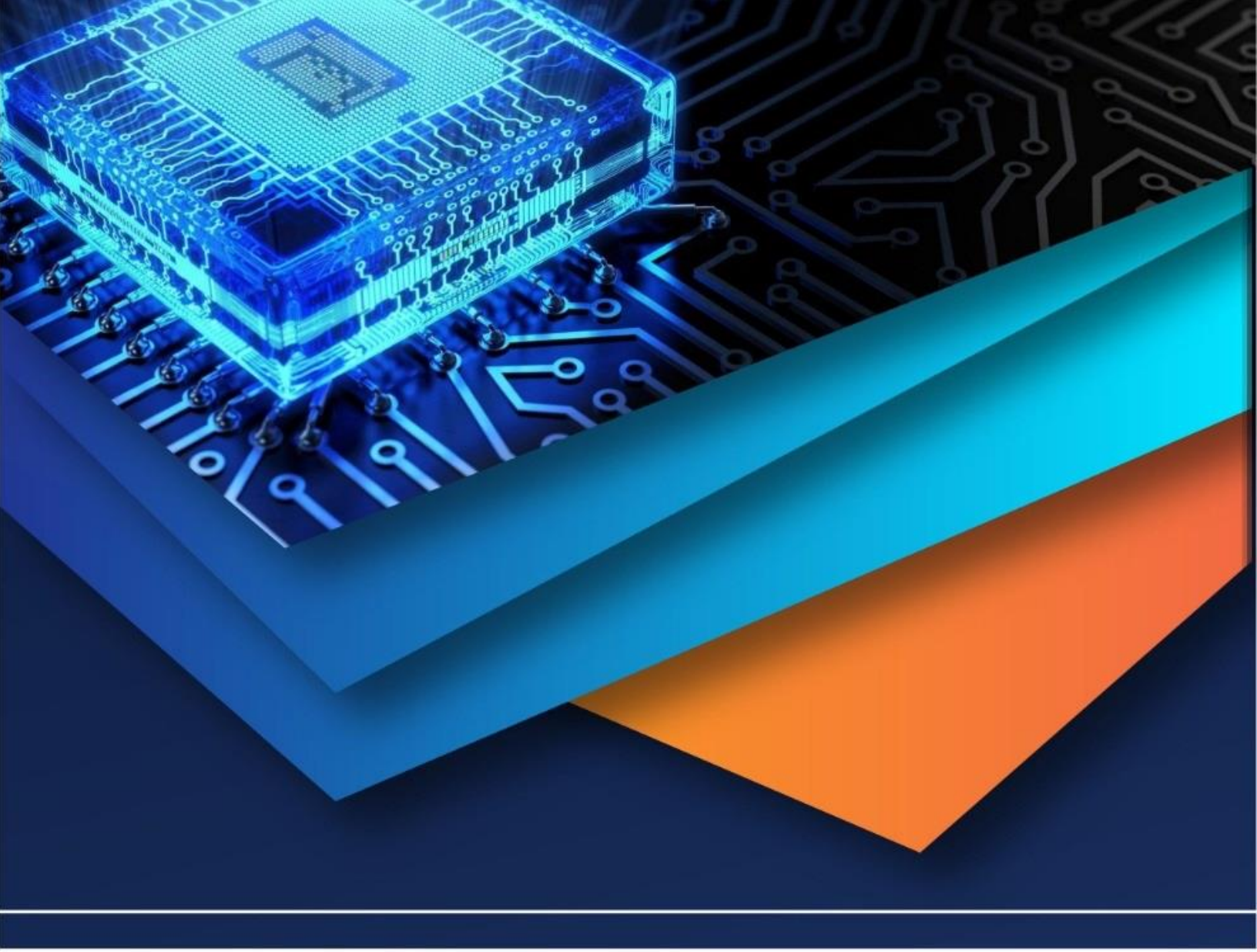
VI. CONCLUSION

In the proposed system, we have implemented an organization-oriented system that would assist the programmer to solve the errors by automatically suggesting the best solution. The system could be used in many business sectors and educational sectors that require strong flawless code, thus reducing the workload of the programmer.

REFERENCES

[1] Maximilian Jenders, Ralf Krestel, Naumann, “Which Answer is Best? Predicting Accepted Answers in MOOC Forums” , conference paper published in 2016.
[2] Ziyu Yao, Daniel S. Weld, Wei-Peng Chen, Huan Sun, “A Systematically Mined Question-Code Dataset from Stack Overflow , conference paper published in 2018.
[3] Chethana V, Evelyn VidyuLatha P, “ KweriME: A Q&A based model which predicts the accepted answers of questions in CQA sites”, journal paper published in 2019.
[4] Ahmad Diyanti, Behrooz Shahi, Syed Mastafa, Mohammed Hadi, Mohammed Hassan Diyanti, “A Proposed approach to determine expertise level of StackOverflow programmers based on mining of user comments” , journal paper published in 2020.
[5] Luca Ponzanelli, Gabriele Bavota, Massimiliano Di Penta, Rocco Oliveto, Michele Lanza, “Prompter: A Self-confident Recommender System” , journal paper published in 2014.
[6] Jun-Wei Lin, Tzu-Chi Lin, Peter Schaedler, “Predicting the Best Answers for Questions on Stack Overflow”, conference paper published in 2016.
[7] Fabio Calefato1, Filippo Lanubile 2 & Nicole Novielli2“An empirical assessment of best-answer prediction models in technical Q&A sites”, conference paper published in 2016.
[8] László Tóth, Balázs Nagy, Dávid Janthó, “Towards an Accurate Prediction of the Question Quality on Stack Overflow Using a Deep-Learning-Based NLP Approach", conference paper published in 2019.
[9] Antoaneta Baltadzhieva, Grzegorz Chrupała, “Predicting the Quality of Questions on Stackoverflow”, journal paper published in 2016.
[10] Amirreza Shirani†, Bowen Xu‡, David Lo‡, Thamar Solorio†, and Amin Alipour†, “Question Relatedness on Stack Overflow: The Task, Dataset, and Corpus-inspired Models ”, conference paper published in 2018.
[11] László Tóth, Balázs Nagy, Dávid Janthó, László Vidács, and Tibor Gyimóthy, “Towards an Accurate Prediction of the Question Quality on Stack Overflow using a Deep-Learning-Based NLP Approach ”, conference paper published in 2018.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)