An Approach to Bug Triage: A Review

Daruthi Rani Chauhan
RSR Rungta College of Engineering and Technology
Dept. of Computer Science and Engineering
Bhilai, Chhattisgarh, India

Prof. Sachin Harne
RSR Rungta College of Engineering and Technology
Dept. of Computer Science and Engineering
Bhilai, Chhattisgarh, India

Abstract

Bugs are one of the important issue for a software organization. According to research, Software organization spends over 45 percentages of their revenue in handling these bugs. To Manage these bugs manually is very difficult and also error prone. Therefore, an automatic approach of instance selection method and feature selection method is combined together to handle the bugs, then the bugs are distributed to bug solving experts. An avoidable step in fixing the bugs is assigning a bug to solving expert. The issue is majority of bugs are assigned to solving experts who has very less experience in that field which can leave the bugs as it is. Therefore using term selection method for a bug solving expert is predicted automatically depending upon which type of bugs it is. A history of these cleared bugs is maintained using historical data management system. There are variety of existing techniques for bug triage. In this paper, we will survey some of these techniques. Most of these techniques provide automatic bug triage and some of these techniques are further classified.

Keywords— *bug data reduction, feature selection technique, instance selection technique, prediction for reduction orders, bug triage.*

I. INTRODUCTION

Bugs are one of the important issue for a software organization. Many of the software companies require to deal with huge amount of software bugs daily. Software bugs are inevitable and fixing these software bugs is a very expensive task. In fact, many of the Software organization spends lots of their resources in handling these bugs. Bug repository are there for managing this bugs. Bugs repository plays an important role. In software development and Maintenance, a bug repository is an important software repository for storing the bugs which has been submitted by the users.

Most of the software which works for an open source projects has an open bug repository system which allows every developers as well as users to submit problems or defects in the software that helps to suggest possible solutions and their remark on existing bug reports. The big drawback is that large-scale software projects are so much huge which makes the triaging process so much difficult. The inefficient data as well as unclear data add redundancy data to the data repository system and create a great problem to the software experts. In bug repository, each software bug has their own bug report. The bug report usually consists of textual information related to the bug and updates regarding to status of bug fixing.

A bug repository provides a data stage support about various types of tasks on bugs, e.g., bug localization, fault prediction, and reopened bug analysis. Huge software projects provides bug repositories which is also called bug tracking systems that support information collection and to assist the developers to handle bugs.

The huge amount of regular occurring bugs for open source big software projects is so much large that makes the triaging process most difficult and challenging. When a bug report is formed, a human triage provides this bug to a developer, who will try to fix this bugs. This developer usually recorded in an item assigned to. The method of providing a correct developer for fixing the bug is known as bug triage. Bug triage basically is one of the most time consuming step in managing of bugs in software projects. Manually bug triage by a human triage is very time consuming process and error-prone therefore the number of daily bugs is huge and lack of knowledge and experience in developers about all bugs. Because of all these things, bug triage outcome in expensive time loss, high cost and lack of accuracy.

Before modifying and verifying a bug, each bug report must be provide to a relevant developer who could fix it. In traditional bug repositories system, all the huge amount of bugs are manually triaged by some specialized developers. The aim is to decrease the human labor costs, some supervised text classification approaches have been proposed for automatic bug triage.

Then the nature of the bugs is predicted by using a predictive algorithm and then prediction of relevant developers for the incoming bug reports with these classifiers.

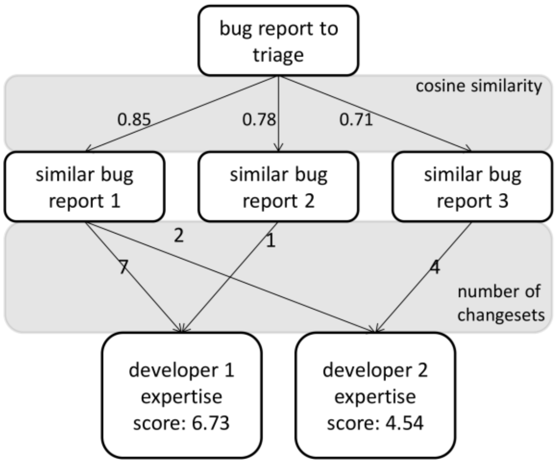

Fig.1. shows basic approach for bug triage

A bug repository provides a data stage support system about types of tasks on bugs, e.g., fault prediction, bug localization, and reopened bug analysis. Huge software projects provide bug repositories which is also called as bug tracking systems which helps to support information collection and to assist developers to handle bugs.

II. LITERATURE SURVEY

B Jifeng Xuan[1], Author focused Towards Effective Troubleshooting with Data Truncation deals with reducing the data present in the bug repository and improve the quality of data then reduce time and cost of bug triaging, it represent an automatic approach to predict a developer with relevant experience to solve the new coming report.. The bug data sets are obtained and techniques such as instance selection feature selection are applied simultaneously.

Suvarnaa Kale [2], focused on A Technique to Combine Feature Selection with Instance Selection for Effective Bug Triage .It addresses the issue of data reduction for bug triage by text classification techniques. Conventional software analysis is not totally suitable for the large-scale and complex data in software repositories. Data mining has developed as a promising means to handle software data. There are two difficulties related to bug data that may influence the effective use of bug repositories in software development tasks, namely the huge scale and the low quality. Therefore unfixed bugs are deleted from the bug repositories.

JifengXuan [3], focused on Automatic Bug Triage using Semi-Supervised Text Classification propose a semi-supervised text classification approach for bug triage to avoid the deficiency of labeled bug reports in existing supervised approaches. This approach combines naive bayes classifier and expectation maximization to take advantage of both labeled and unlabeled bug reports. This approach trains a classifier with a fraction of labeled bug reports. Then the approach iteratively labels numerous unlabeled bug reports and trains a new classifier with labels of all the bug reports. Then it employs a weighted recommendation list to boost the performance by imposing the weights of multiple developers in training the classifier. Before training a supervised classifier for bug triage, a necessary step is to collect numerous labeled bug reports, which are bug reports marked with their relevant developers. The semi supervised text classification approach to improve the classification accuracy of bug triage. This semi supervised approach enhances a NB classifier by applying expectation-maximization (EM) based on the combination of unlabeled and labeled bug reports. First, this semi-supervised approach trains a classifier with labeled bug reports. Then, the approach iteratively labels the unlabeled bug reports and trains a new classifier with labels of all the bug reports.

Phuc Nhan Minh [4], focused on An Approach to Detecting Duplicate Bug Reports using N-gram Features and Cluster Chrinkage Technique to duplication detection which is an important issue for software maintenance in recent years. In this study, we propose a detection scheme using n-gram features and the cluster shrinkage technique. From the empirical experiments on three open source software projects, the proposed scheme shows its effectiveness in duplication detection.

Anjali [5], focused on Bug Triaging: Profile Oriented Developer Recommendation. Author proposed a Domain Mapping Matrix (DMM) based developer recommendation approach for predicting the best suited developers list who could resolve the newly reported bugs. Unlike other approaches, our approach does not find a matching with historical bug reports and recommend the developers who fixed historical bug report; rather, it utilizes the expertise profile of developers maintained in DMM. This profile can be easily updated with time. Using the developer profile is better as the number of token matching is more. For any new bug report all of its tokens are matched in the dataset and expertise list corresponding to new bug report tokens is generated. Thus, our proposed approach uses a wider area for token matching.

***TABLE 1**. Shows comparison between various existing approaches and its limitation*

Reference	Method Used	Data Source	Approach	Strength	Limitation
[1]	Instance selection and feature selection	Mozilla and Eclipse Bug	In this paper, author combine feature selection with instance selection to reduce the scale of bug data sets as well as improve the data quality	An approach to leveraging techniques on data processing to form reduced and high-quality bug data in software development and maintenance.	Need to prepare a high quality bug data set and tackle a domain-specific software task.
[2]	Instance selection and feature selection	Mozilla and Eclipse Bug	Paper gives a way to deal with utilizing methods on data processing to form reduced and high-quality bug data in software development and maintenance and also improve the results of data reduction in bug triage.	It analysis data by considering the word dimension and bug dimension which helps in reducing duplicate and unnecessary bugs	The order of applying instance selection and feature selection is not clearly explained which leads to inefficient system.
[3]	Naïve Bayes classifier	Eclipse and Net beans Bug	Author propose a combination of naive Bayes classifier and expectation maximization to take advantage of both labeled and unlabeled bug reports	It labels the bug data iteratively. The weighted list maintained, helps to boost the results obtained.	It only focuses on classifying the bugs in bug repository. The major problem in bug handling is that huge number of data in bug repository.
[4]	N-gram Features, Cluster Chrinkage Technique	Eclipse Bug	Proposed approach has two novel features: firstly, use n-gram features for the task of duplicate bug report detection. Secondly, apply cluster shrinkage technique to improve the detection performance.	The technique has provided to improve the Detecting Duplicate Bug performance.	This technique is applicable for only AgroUML, Apache and SVN and might not applicable for Eclipse and Firefox
[5]	Domain Mapping Matrix (DMM)	chromium bug repository	Propose a new approach for selecting the developers who have appropriate expertise in the related area for handling the bug reports.	Rather than using historical bug report it uses domain mapping matrix for expertise profile of developer maintenance.	No practical dataset tested

III. CONCLUSION

The huge amount of data available in the bug repository which helps to play an important role in bug handling. Therefore, as to decrease the data in bug repository, the bug reduction approaches must be implemented. The data in bug repository is mainly decreased by neglecting the redundancy of data in the bug repository. Then, the subset of the data in data repository is acquired. In future we can to develop a predictive model which helps to predict a developer on the basis of the type of bugs obtained.

REFERENCES

[1] B Jifeng Xuan, He Jiang, Yan Hu, Zhilei Ren, Weiqin Zou, Zhongxuan Luo, and Xindong Wu," Towards Effective Bug Triage with Software Data Reduction Techniques" ieee transactions on knowledge and data engineering, vol. 27, no. 1, january 2015.

[2] Suvarnaa Kale, Ajay Kumar Gupta, "A Technique to Combine Feature Selection with Instance Selection for Effective Bug Triage", "International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064".

[3] D. Cubranic and G. C. Murphy , "Automatic bugs triaging usings text categorization," in Proc Sixteenth International Conference on Software Engineering, Citeseer, 2004, pp. 92–97.

[4] Phuc Nhan Minh, "An Approach to Detecting Duplicate Bug Reports using Ngram Features and Cluster Chrinkage Technique", International Journal of Scientific and Research Publications, Volume 4, Issue 5, May 2014, ISSN 2250-3153.

[5] Anjali, Sandeep Kumar Singh, "Bug Triaging: Profile Oriented Developer Recommendation", International Journal of Innovative Research in Advanced Engineering (IJIRAE), ISSN: 2349-2163, Volume 2 Issue 1 (January 2015).

[6] M.Suresh, M.Amarnath, G.Baranikumar, M. Jagadheeswaran, "A SURVEY ON BUG TRACKING SYSTEM FOR EFFECTIVE BUG CLEARANCE", International Research Journal of Engineering and Technology (IRJET), Volume: 03 Issue: 02, pp.- 1622-1630, Feb-2016