© 2016 IJSRST | Volume 2 | Issue 2 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X
Themed Section: Science and Technology

Software Maintenance and Resolving the Bugs for Bug Triage

R. Sneha, G. Subhashree, M. Murugesan

Department of Computer Science and Engineering, Dhanalakshmi College of Engineering, Manimangalam, Tambaram, Chennai, Tamil Nadu, India

ABSTRACT

Fixing bugs in software development companies is a big problem which needs a lot of money. A bug repository manages the software bugs. Usually the software projects include many developers. The duty of the project manager is to assign projects to the developers. In the existing system we use text classification technique for assigning projects or bugs which consumes a large space and produces a low quality bug data. To overcome this problem, instance selection and feature selection is introduce to increase accuracy and decrease scale all together. In our work we combine both these techniques in bug repositories to reduce data.

Keywords: Preprocessing of data, managing bugs, instance selection, feature selection, bug triage, bug reduction orders, bug fixing.

I. INTRODUCTION

In current world, software companies maintains a large-scale databases for storing the output of the software project, e.g., source code, bugs, emails, and specifications. Conventional software analysis is not fully suitable for the large-scale and complex data in bug repositories. Hence, Data mining has a promising means to handle bugs. Mining repositories can uncover interesting information in software repositories and solve the real world software problems. A bug repository has a collection of bug repots that plays an important role in managing bugs. Software bugs are always happening and clearing bugs is costly in software development. Huge software projects maintain bug repositories to collect the information and to help developers to handle the bugs. A bug repository maintains a bug report, which records the description the bug and the updates about the status of bug that has to be fixed. A bug repository contains several types of bugs such as fault prediction, bug localization, and reopened bug analysis.

II. METHODS AND MATERIAL

A. Existing Work

In traditional software development, bugs are triaged by human triager, the developer who triages the new bugs manually. Triaging huge number of bugs manually takes more time and cost. To overcome this problem, an automatic bug triage system is introduced in existing system. It uses text classification technique, in which each reported bug is assigned to a developer. Developer is mapped to the label of the document containing bugs that are to be resolved. Bug triage is then converted into a problem of text classification and bugs are automatically solved with text classification techniques, e.g., Naive Bayes. From the results of text classification, a human triager assigns new bugs by incorporating his/her expertise. In text classification techniques accuracy can be increased by investigating some further techniques e.g., a tossing graph approach and a collaborative filtering approach. However, the techniques of automatic bug triage are blocked in bug repositories which are low in quality. As data are a kind of simple text data, the well-processed bug data has to be generated to facilitate the application.

B. Proposed Scheme

In proposed system, the problem of data reduction for bug triage is addressed, i.e., how the bug data is reduced to bring down the labor cost of developers and the quality is improved to facilitate the process of bug triage. Data is reduced by removing bug reports and words, which are redundant or non-informative. Bug triage aims to build an efficient set of bug data .In our work, the bug dimension and the word dimension are reduced simultaneously by combining the techniques instance selection and feature selection. Thus the reduced bug data has lesser number of bug reports and lesser number of words than the original bug data. Although data is reduced it provides similar information as it is in the original bug data. The results of four instance selection algorithms and four feature selection algorithms are examined to avoid the bias of a single algorithm .when an instance selection algorithm and a feature selection algorithm is given, the order of applying these two algorithms may affect the results of bug triage.

In the proposed paper, predictive model is used to determine the order of applying instance selection and feature selection. From the experiments conducted over bug repots, it is identified that applying instance selection technique to the data set can reduce relevant subset of bug reports but the accuracy of bug triage may be decreased; applying the feature selection technique can reduce subset of relevant words in the bug data and the accuracy can be increased. Hence it is found that combining both these techniques can increase the accuracy, as well as reduce bug reports and words.

Contributions of this paper are as follows:

a) To simultaneously reduce the scales of the bug dimension and the word dimension
b) To improve the accuracy of bug triage.
c) Combination approach is proposed to address the problem of data reduction. That is application of instance selection and feature selection in bug repositories.
d) A binary classifier is built to predict the order of applying instance selection and feature selection.

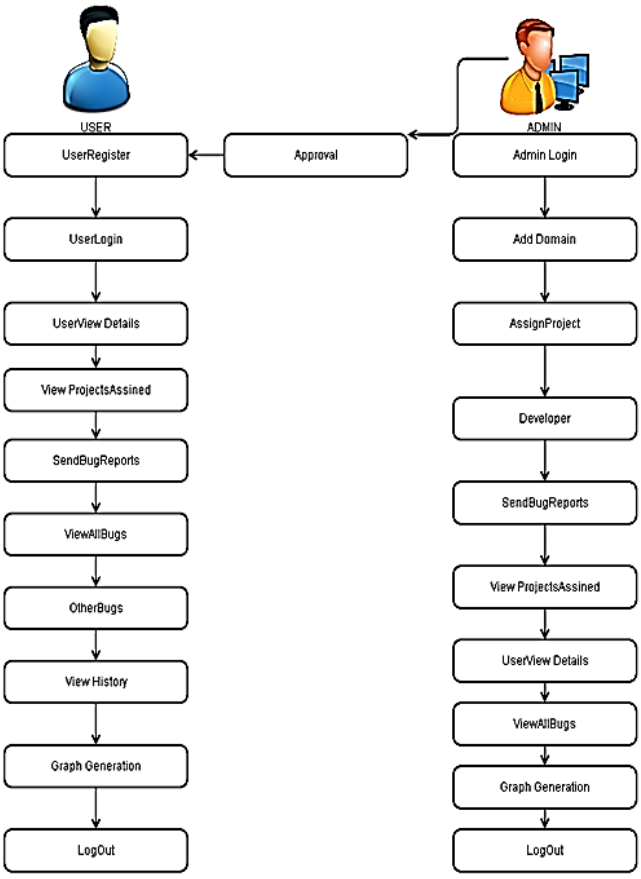

Figure 1. Proposed System

C. Background And Motivation Bug Triage

A bug tiage is a formal process where each bug is priotitized based on its severity, frequency, risk and etc. The bugs are sorted based on the priority. Bugs with high priority will be fixed first. Low priority bugs will be fixed later. To achieve better balance working, important bugs are prioritized first. Three ways of differentiating bugs:

a) bug which is fixed.
b) bug which is fixed later and
c) bug which is not fixed.

A bug triage meeting for developers is held regularly for discussing about project life cycle. The Quality Assurance lead calls these meetings. The number of occurrences of meeting will vary from project to project, based on current status of project.

Motivation

Real-world data consists of noisy and redundant data which leads to increase in the cost factors of data processing. In the bug repository, all the bug reports are filled by developers in normal language. The bugs which are low in quality are collected in the bug repositories with the growth in scale. Such data which are inefficient may worsen the accuracy of fixing bugs.

III. RESULT AND DISCUSSION

A. Experimental Work

Data preparation

Data preparation means manipulating data into a form suitable for further analysis. Data preparation is the process of collecting, cleaning, and consolidating data. It is a process of pre-processing data.

Here, we present the data preparation for applying the bug data reduction. Bug triage predicts developers to fix bugs. The unfixed bugs are assigned to a developer to fix it. In bug repositories, some registered developers may have fixed very less number of bugs. Such in active developers who fixed less than 10 bugs are removed. Applying Instance Selection and Feature Selection.

In our work, we combine instance selection and feature selection to perform data reduction with more accuracy. The original data set is replaced with the reduced data set for bug triage. Instance selection and feature selection are widely used techniques in data processing. Instance selection is to obtain a subset of relevant instances (i.e., bug reports in bug data) while feature selection is to obtain a subset of relevant features (i.e., words in bug data). Instance selection technique reduces the number of instances by removing noisy and redundant instances. An instance selection algorithm can provide a reduced data set by removing relevant instances. Feature selection is a pre-processing technique for reducing relevant features for big scale data sets. When an instance selection algorithm IS and a feature selection algorithm FS is given, the order of applying has to be predicted. We use FS->IS to denote the bug data reduction, which first applies FS and then IS, and in IS->FS first applies IS and then FS.

B. Benefits of Data Reduction

Reducing the Data Scale

Bug dimension: Bug triage is to assign developers for bug fixing. Developers can examine history of fixed bugs to find a solution to the current bug report. If the bug is already fixed then it can be replaced. The labor cost of developers can be saved by fixing bugs from history using instance selection, instead fixing in their own manually. The reduced data set can be handled more easily by automatic techniques (e.g., bug triage approaches) than the original data set.

Improving the Accuracy

Bug dimension: Instance selection can remove uninformative bug reports, while the accuracy may be decreased by removing bug reports. Word dimension: feature selection removes uninformative words, but the accuracy of bug triage is improved. This can recover the accuracy loss by instance selection.

IV. CONCLUSION

A bug triage is a process done by software companies in order to maintain the bugs in their work process. In this project we use techniques such as instance selection and feature selection in order to improve the efficiency and maintenance of the bug. Thus the redundancy in the bug data set will be removed. It also helps us to assign the correct the correct project to the correct developer .The developer fixes less than 10 bugs will be removed by the Quality analyst (QA) during the bug triage process meet.

V. REFERENCES

[1] Y. Yang and J. Pedersen, "A comparative study on feature selection in text categorization, "in Proc. Int. Conf. Mach.Learn, 1997, pp. 412–420.

[2] H. Zhang and G. Sun, "Optimal reference subset selection for nearest neighbor classification by tabu search," Pattern Recognit., vol. 35 pp. 1481–1490, 2002.

[3] Y. Yang, "An evaluation of statistical approaches to text categorization," Inform. Retrieval, vol. 1, pp. 69–90, 1999.

[4] R. S. Pressman, Software Engineering: A Practitioner's Approach, 7th ed. New York, USA: McGraw-Hill, 2010.

[5] "Towards Effective Bug Triage with Software Data Reduction Techniques", Jifeng Xuan, He Jiang, Member, IEEE, Yan Hu, Zhilei Ren, Weiqin Zou, Zhongxuan Luo, and Xindong Wu, Fellow, IEEE, TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING,VOL. 27, jan 2015.