Scientific Journal Impact Factor (SJIF) : 1.711
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745 ISSN (PRINT) : 2393 - 8161

A Literature Review on Bug Triage with Software Data Reduction Techniques

Miss. Tejaswini Pratap Aher[1], Prof. M. S. Ankoshe[2]

[1] *M.E. Computer, Department of Computer Engineering, SRES's COE ,Kopargaon, Maharashtra , India*

[2] *Assistant Professor, Department of Computer Engineering, SRES's COE ,Kopargaon, Maharashtra , India*

Abstract— This work proposes a literature survey on Bug Triage with Software Data Reduction Technique. To take step of fixing bugs in bug triage, which aims to correctly assign a developer to a new bug. Large software companies spend large cost software bugs. To decrease the time cost in manual work, text classification techniques are applied to conduct automatic bug triage. Here, we address the problem of data reduction for bug triage and strategies to improve the instance selection with feature selection to simultaneously reduce data scale on the bug dimension and the word dimension. We extract attributes from historical bug data sets and build a predictive model for a new bug data set. The results show that our data reduction can reduce the data scale and improve the accuracy of bug triage. This work provides an approaches on data processing to form reduced and high-quality bug data in software development and maintenance.

Keywords— Bug repositories, Bug triage, Bug data reduction, Feature selection, Instance selection.

I.INTRODUCTION

Software repositories help in data mining to solve the problems of software engineering. Software repositories contain the source code, bugs, mail and many other specifications. This output is stored in large software repositories. Data mining has emerged to solve the problem of large software analysis and complexities in bug repositories. Data mining also helps to uncover the information in software repositories.

Once a software bug is found, the bug gets reported to the bug repository. Once a bug report is formed, a human triage assigns that bug to a person who will try to fix the bug who is a developer .This developer is recorded in an item assigned-to .If the previously assigned developer cannot fix this bug, the assigned-to will change to another developer. This process of assigning a correct developer for fixing the bug is called as a bug triage. A developer starts to fix the bug based on the knowledge of historical bug fixing. Again developer has to pay efforts to understand the new bug report and examine historically fixed bugs. We have many existing approaches for text classification to study the bug triage where the description of the bug report is extracted. The developer fix this bug and mark the accuracy.

- **Bug Triage Approach**

Here the bug triages approach, helps developers by applying various text classification techniques. In this approach, a bug report is mapped to a document and a then developer is mapped to the label of that document. Bug triage is then converted into a problem of text classification and is automatically solved with mature text classification techniques, e.g., Naive Bayes. Based in the result of text classification, human triage assigns new bugs by her or his expertise. To improve the quality to facilitate the process of bug triage and to reduce the bug data to save the labor cost we address the problem of data reduction for bug triage. Data reduction helps bug triage by removing non-informative and redundant information. So, here we are in need to combine the existing techniques of instance selection and feature selection combine existing technique reduces the bug dimension and word dimensions. This reduced bug data contain fewer bug reports and fewer words than the original

bug data and provide similar information over the original bug data. We evaluate the reduced bug data according to two criteria: the scale of a data set and the accuracy of bug triage.

II. LITERATURE REVIEW

Sandusky, Gasser and Ripoche gave the feature of the defect tracking repositories, which showed the evolvement of the bug report network which shows formal and informal relationship in the bug data and also to examine the dependency among the bug report[1].

Q. Hong, S. Kim et.al gave theory for understanding large software development and maintenance demand participation of group of developers for improving development and maintenance quality and reduction cost[2]. J. Xuan, H. Jiang et.al identified the developer prioritization which can distinguish developers and assists task in software maintenance [3].

Thomas Zimmermann and Silvia Breu suggested that many follow-up questions are needed to be posed to the reporters of bug. So, there is high need of efficient and effective communication with the teams in open source project. These gave the high results of bug fixing activities and have updates of the bug. Integration and active participation of users in bug tracking will result in efficient bug tracking.[10].

The software development, bug provides vital information to developer. But, they differ in quality. Zimmerman and Battenberg identified the quality of bug data, the designing questionnaires to developers and users. Based on this questionnaires, we can characterize what makes a good bug report and what classifier are to be trained to identify the quality of bug and strategies to improve them[4].

To detect the duplicate bug report which weakens the quality of bug, Sun and Jaing described duplicate bug detection approach by optimizing a retrieval function on multiple features [5].

D. Cubranic and G. C. Murphy solved the problem of assigning text document into one or more categories or classes, we apply the text classification on this [6].

J. Anvik and G. C. Murphy recommended applying bug triage which improves the software development process. A triage determines, if the report are meaningful, these report are then organized for integration of the project's development process [7].

A. Lamkanfi, S. Demeyer predicted that the severity of reported bug is a critical factor and decides that soon it needs to be fixed. We can do this by textual description using text-mining algorithms[8].

Rogati and Yang reported a four well-known classification algorithms a Naïve Bayesian(NB) approach,a Rocchio-style classifier,a k-Nearest Neighbour(kNN) method and a Support Vector Machine(SVM) system. This gave analysis of four classification algorithms and mapped the performance measures[11].

Above result gave that the Naïve Bayes classification improves with smaller training set. The data set was divided into a test set and train set by randomly selecting a percentage of a documents from the data set. The data set are to be placed into train set and classification is performed.

We address the problem of data reduction for bug triage. As per our knowledge, we have no existing work which has investigated the bug data sets for bug triage. In contrast to bug triage, defect prediction is a binary classification problem, which aims to predict a software artifact contains flaws according to extracted features. So, T. M. Khoshagotkar, K. Gao, N. Seliya examine the techniques on feature selection to handle imbalanced data[9].

III. CONCLUSION

Thus, above literature survey concludes that, no any author has gone for applying data reduction, reduce data scale and improve the accuracy of bug triage. Any interested researcher may combine instance selection and feature selection to reduce data scale on the bug dimension and the word dimension. A researcher can extract attributes form historical bug data set and build a predictive model for new bug set, to determine the order of applying instance selection and feature selection.

@IJMTER-2016, All rights Reserved

IV. ACKNOWLEDGEMENT

I express my sincere gratitude to Prof. D. B. Kshirsagar, Head of Department (Computer), Prof. P. N. Kalavadekar, (PG Co-ordinator) and Prof. M. S. Ankoshe (Guide) SRESCOE for his unending support and encouragement during the years I have studied under his tutelage.

REFERENCES

I. R. J. Sandusky, L. Gasser, and G. Ripoche, "Bug report networks: Varieties, strategies, and impacts in an F/OSS development community," in Proc. 1st Intl. Workshop Mining Softw. Repositories, May 2004.
II. Q. Hong, S. Kim, S. C. Cheung, and C. Bird, "Understanding a developer social network and its evolution," in Proc. 27th IEEE Int. Conf. Softw. Maintenance, Sep. 2011.
III. J. Xuan, H. Jiang, Z. Ren, and W. Zou, "Developer prioritization in bug repositories," in Proc. 34th Int. Conf. Softw. Eng., 2012.
IV. T. Zimmermann, R. Premraj, N. Bettenburg, S. Just, A. Schr€oter, and C. Weiss, "What makes a good bug report?" IEEE Trans. Softw. Eng., vol. 36, no. 5, Oct. 2010.
V. C. Sun, D. Lo, S. C. Khoo, and J. Jiang, "Towards more accurate retrieval of duplicate bug reports," in Proc. 26th IEEE/ACM Int. Conf. Automated Softw. Eng., 2011.
VI. D. Cubrani_c and G. C. Murphy, "Automatic bug triage using text categorization," in Proc. 16th Int. Conf. Softw. Eng. Knowl. Eng., Jun. 2004.
VII. J. Anvik and G. C. Murphy, "Reducing the effort of bug report triage: Recommenders for development-oriented decisions," ACM Trans. Soft. Eng. Methodol., vol. 20, no. 3, article 10, Aug. 2011.
VIII. A. Lamkanfi, S. Demeyer, E. Giger, and B. Goethals, "Predicting the severity of a reported bug," in Proc. 7th IEEE Working Conf. Mining Softw. Repositories, May 2010.
IX. T. M. Khoshgoftaar, K. Gao, and N. Seliya, "Attribute selection and imbalanced data: Problems in software defect prediction," in Proc. 22nd IEEE Int. Conf. Tools Artif. Intell., Oct. 2010.
X. S. Breu, R. Premraj, J. Sillito, and T. Zimmermann, "Information needs in bug reports: Improving cooperation between developers and users," in Proc. ACM Conf. Comput. Supported Cooperative Work, Feb. 2010.
XI. M. Rogati and Y. Yang, "High-performing feature selection for text classification," in Proc. 11th Int. Conf. Inform. Knowl. Manag., Nov. 2002.
XII. D.Cubrani_c and G. C. Murphy, "Automatic bug triage using text categorization," in Proc. 16th Int. Conf. Softw. Eng. Knowl. Eng., Jun. 2004.

@IJMTER-2016, All rights Reserved