International Journal of Engineering and Techniques - Volume 2 Issue 6, Nov – Dec 2016

RESEARCH ARTICLE OPEN ACCESS

Automatic Bug Triage with Data Reduction

Fareen Sayed, Hetvi Shah, Prajakta Ohal, Namrata Kharat, Gopal Deshmukh

[1,2,3,4,5](Department of Computer Engineering, MESCOE, Pune)

Abstract:

Bug triage means to transfer a new bug to expertise developer. The manual bug triage is opulent in time and poor in accuracy, there is a need to automatize the bug triage process. In order to automate the bug triage process, text classification techniques are applied using stopword removal and stemming. In our proposed work we have used NB-Classifiers to predict the developers. The data reduction techniques like instance selection and keyword selection are used to obtain bug report and words. This will help the system to predict only those developers who are expertise in solving the assigned bug. We will also provide the change of status of bug report i.e. if the bug is solved then the bug report will be updated. If a particular developer fails to solve the bug then the bug will go back to another developer.

Keywords — **Bug triage, Instance Selection, Keyword Selection, Bug Report, Data Reduction, Text Classification.**

I. INTRODUCTION

Every software company approximately deals with bugs in all sorts of projects. Software bugs leads to poor user skills and low throughput. A bug repository plays a vital role in handling the software bugs. A bug repository maintains a bug report that contains the description and status of the bug fixing.

In open source projects, an average of 30 to 100 bugs is disclosed per day. In expansion of any software structure, software change demands are often generated. Most of these changes are mutual to errors created in programs. Also it is necessary for a software developer to decide how rapidly bugs will get fixed. A bug repository may sometimes contain bugs that are unseen and unsolved. Due to this the quality of software may decrease. To improve the quality of software being developed, it is necessary to constantly keep a track of bug report and assign each unsolved bugs to expertise developer. In classic software development, mechanism of assigning bugs is manual. In this approach, the developer reads each bug in detail and then decides whether to solve that bug or not. Also, many times the bug is assigned to developer who isn't capable of solving that bug. This process is time consuming for huge projects where bug list grow at fast rate. In every software organizations, for boosting their production quality it is important to automate the bug triage process.

To minimize the amount of time and expense in manual work, text classification methods are enforced to for bug triage process. Data reduction is used to determine how to reduce the scale and enhance the data quality. In this paper, we combine keyword selection and instance selection algorithm, in order to reduce bug dimension and word dimension. In our work, the prediction of developers will depend on his/her historical data or the information provided by the developers during the registration process.

The remainder of the paper is organized as follows. Section II provides Literature Review. Section III presents the background. Section IV gives the structure of proposed system. Section V shows the system architecture. Section VI concludes.

II. LITERATURE REVIEW

Jiefng Xuan et al.[1] mark the issue of noisy and low quality of data. They employ data reduction strategies by combining instance selection and feature selection. They made use of predictive model to decide the order of instance selection and feature selection. They used text classification techniques and solved the bugs. In this paper, they have used Naïve Bayes algorithm to predict developers.

[2] demonstrated that it is possible to predict the severity based on the other information in a bug report. They estimated the performance of predictors based on three cases drawn from open-source community.

[3] they tried to improve the project administration.

Based on empirical studies of three CA maintenance projects, they applied they procedure to other software systems to improve software maintenance process.

[4] in this paper, they dealt with data reduction problems. They combined feature selection with instance selection for data diminution to construct the reduced data set and improve the quality of bug data.

[5]in this paper, they have presented a comparative interpretation of various available information retrieval and machine learning methods. They have downloaded resolved bug reports along with the developer activity data from Mozilla open source project. They acquired a combination of methods which was most satisfactory to establish a high performance bug triage system.

[6] proposed the concept for text representation and processing. They made used of distance graph to represent the documents in terms of distance between the distinct words. This provided a much richer representation in terms of sentence structure of the underlying data.

III. BACKGROUND

A software bug is an error or a failure in a program or system that leads it to produce flawed or abrupt outcomes. Today, users of software organizations are inspired to report the bugs they confront, using bug tracking systems like Bugzilla, Mantis, Jira and Trac. For a bug tracking system, it is necessary to have information about the bug, history or work log, tools which can be used to store, retrieve and organize bug information.

Once a bug is established, a reporter (normally a developer, a tester or an end user) reports a bug into bug repository. All information of a bug is recorded in a bug report which contains various items (such as assigned-to, history, status, etc.) for recreating the bug. A bug report contains two elements, namely summary and depiction, which are recorded in text format. A general explanation for analyzing a bug is expressed by summary while the depiction gives the specifications for reproducing the bug. After a bug report is formed, a human tracker will assign this bug to developer, who will fix the bug. The technique of assigning a perfect developer for repairing the bug is called bug triage. Assigned developer starts to fix the bug, based on the history of bug. A bug report contains an item status, which is changed according to present outcome until the bug is absolutely fixed.

IV. PROPOSED SYSTEM

In traditional software establishment, new bugs are human triaged by a skilled developer. This approach has disadvantages, as there are large number of bugs and a shortage of expertise to fix all bugs. Also, it has poor accuracy and is time consuming. To desist from high cost of manual triage, an automatic way for bug triage process is proposed.

Fig. 1, we illustrate the basic frame work of bug triage based on text classification. A bug data set contains bug reports with respective developers. On this bug data set, the bug data reduction is applied as a phase in data preparation of bug triage. Work combines existing techniques of instance selection and keyword (feature) selection to remove certain bug reports and words. A problem for reducing the bug data is to discover the order of instance selection and keyword selection, which is denoted as the prediction of reduction orders.

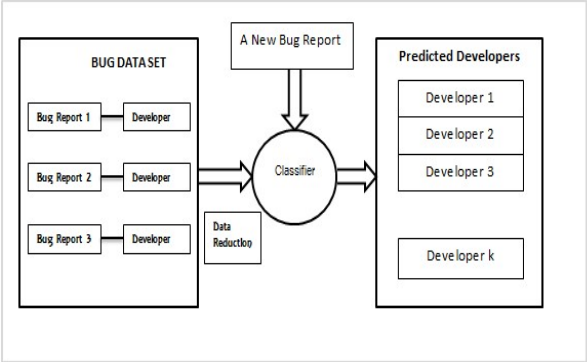

Fig. 1 Framework of Bug Triage Based On Text Classification.

In bug triage, a bug data set is converted into text matrix with two dimensions, namely the bug dimension and the word dimension. Work leverages the combination of instance selection and keyword selection to generate a reduced bug data set. We replace the original data set with the reduced data set for bug triage. For given data set in a certain application, instance selection is to obtain a subset of suitable instances while keyword selection aims to obtain a subset of suitable words in bug data set. Given an instance selection algorithm IS and keyword selection algorithm KS, we use KS!IS to denote the bug data reduction, which first applies KS and then IS; on the other hand, IS!KS denotes first applying IS and then KS. We briefly present how to reduce the bug data based on KS!IS. Two algorithms KS and IS are applied sequentially.

V. SYSTEM ARCHITECTURE

A. *Bug Repository And Bug Record*

A bug repository is a common software storehouse for storing characteristics of bugs.In bug repository, a bug is preserved as a record, which reports textual explaination for replicating the bug and renovates corresponding to the status of bug mending.

A. TEXT CLASSIFICATION

Text classification allots one or more classes to document based on their content.

a) Stopword Removal

Natural languages frequently make use of productive terms, such as verbs,conjunctions,adverbs and preposition to frame up sentences. Words which don't import peculiar knowledge in bug report are known as stopwords.For example, 'the','that','and','this',etc.

b) Stemming

Each term arriving in the depiction is reduced into its primary form by stemming process.For example, words 'playing','playful','played' all share the equivqlent semantic base – 'play'.

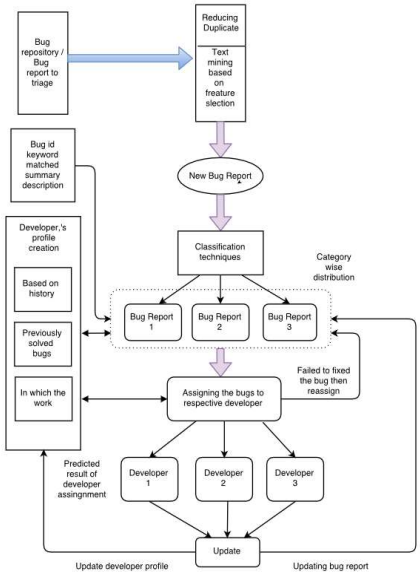

Fig. 2 System Architecture

B. DATA REDUCTION

To prevent the effort cost of developers , we apply data reduction for bug triage. We merge current methods of instance selection and keyword selection discard certain bug reports and words.

a) Instance Selection

Instance selection is a method to decrease the amount of instances by discarding noisy and unwanted instances.For a given data set, instance selection is to retrieve a group of related instances.

b) Keyword Selection

Keyword selection intends to retrieve a group of important words in bug data. It is used for choosing a reduced set of words for large scale data sets.

C. NB- CLASSIFIER

NB classifier algorithm is used to predict developers. The input to this is a bug report and it will predict the topmost developers based on their history.

VI. CONCLUSION

For almost all software system, bug-fixing is an essential scheme. To measure maintenance effort and advance project administration, it is necessary to estimate bug-fixing time. By combining instance selection with keyword selection, we try to trim bug data set as well as enhance bug data quality. In this paper, we obtain characteristics of each bug data set, in-order to determine the sequence of applying instance selection and keyword selection for a new bug. Our objective is to discover a combination of orders, which is most convenient to ultimately develop a powerful bug triage system. We will observe the results on Bugzilla, which is a testing tool developed by Mozilla project.

In future work, we plan improving the results of data reduction in bug triage. Also, we plan to take efforts for solving time constraint problems arising in the bug triage approach.

REFERENCES

1. JifengXuan, He Jiang. Yan Hu, ZhileiRen, WeiqinZou, ZhongxuanLuo, and Xindong Wu,"Towards Effective Bug Triage with

Software Data Reduction Techniques" IEEE Transaction on Knowledge and Data Engineering, vol. 27, no. 1,January 2015.

2. 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.

3. H. Zhang, L. Gong, and S. Versteeg, "Predicting bug-fixing time: An empirical study of commercial software projects," in Proc. 35th Int. Conf. Softw. Eng.,May 2013.

4. Prof. A. Gadekar, N. Waghmare, P. Taralkar, R. Dapke, "Detecting and Reporting bugs by using Data Reduced technique", August 2015.

5. S. N. Ashan, J. Ferzund and F. Wotawa, "Automatic Software Bug Triage System (BTS) Based on Latent Semantic Indexing and Support Vector Machine",
2009.

6. W. Zou, Y. Hu, J. Xuan, and H. Jiang, "Towards training set reduction for bug triage," in Proc. IEEE 35th Annual CS and Application Conference, Washington, DC, USA: IEEE Computer Society, 2011.

7. E. Murphy-Hill, T. Zimmermann, C. Bird, and N. Nagappan, "The design of bug fixes," in Proc. Int. Conf. Softw. Eng., 2013, pp. 332–341.

8. J. Anvik,"Automatic bug report assignment ," in Proc 28th International Conference on Software Engineering. ACM, 2006.

9. C. C. Aggarwal and P. Zhao, "Towards graphical models for text processing",Knowl. Inform. Syst., vol. 36, no. 1, pp.1-21, July 2012.s

10. X. Zhu and X. Wu, "Cost-constrained data acquisition for intelligent data preparation," IEEE Trans. Knowl. Data Eng., vol. 17, no. 11, pp. 1542–1556, Nov. 2005.