iJRASET
International Journal For Research in
Applied Science and Engineering Technology

Volume: 5 Issue: VII Month of publication: July 2017

DOI:

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue VIII, July 2017- Available at www.ijraset.com

An Effective Bug Triage System Using Data Reduction Techniques

Tanuja Pansambal[1], Prof S.P.Vidhate[2]
[1,]Department of Computer Engineering, VACOEA, Savitribai Phule University,Pune
[2]HOD, Department of Computer Engineering, VACOEA, Savitribai Phule University,Pune

Abstract : Fixing of bug consumes more time in Software industry. Bug triage is a way to assign new bug to an appropriate developer. Software Company manually handles bug fixing. Automatic bug assignment is needed for efficient bug triage, which handles large bug data with low quality to get efficient high quality bug data. In this paper, combine feature selection and Instance selection methods for minimizing the bug dimension and word dimension. Extraction of various attributes from previous bug track record is carried out to build new bug track or dataset. The proposed system works on sample data set to generate small amount of but highly informative bug report data. Proposed system also provides graphical analysis for bug triage, which gives employee wise and bug wise analysis. This system Result shows appropriate bug assignment to developer. Proposed system minimizes manual work of bug fixer. Proposed system also analyses performance of tester using bug triage history.

***Keywords**: Instance selection, Feature selection, application of data preprocessing, Mining software repositories, bug data minimization, bug triage, data management for bug repositories, prediction for reduction orders.*

I. INTRODUCTION

Mining various software repositories in these days has become essential and hence employment of data mining tools for mining data for software maintenance is necessary. In the current software development environment, the output of software development is stored into large software repositories for maintaining large scale bug data, e.g. codes, emails, specifications and bugs. Conventional was of software analysis is not convenient for complex and huge data. Software data has been efficiently handled by the data mining techniques in current digital age. In proposed approach for bug data reduction, we scale out the data by analysing and categorizing the bug data which will indirectly increase the quality of the data. Proposed approach is a combinational process of feature selection along with instance selection simultaneously.

Also the proposed system comprised of a novel module to determine the status of the big checking the current state of the bug whether it's rectified or not or it's assigned to some developer or not. Software bug management is carried out with an important entity names as bug repository. An open bug repository is available in many open source software, which not only allow users to report about the issues or defects found in the software, but also suggest future enhancements as per user's needs, and also comment or resolve the existing bug reports. The numbers of bug reports have exponentially increased which is making bug triaging for popular open source software. Two major challenges being faced in software bug repositories are huge amount of bug report data and the quality of bug report. A bug repository maintains as a bug report in a textual description form and is updated according to the status hierarchy of the bug fixing.

II. LITERATURE SURVEY

A. Towards Effective Bug Triage with Software Data Reduction Techniques

In this paper section, we address the problem of information decrease for bug triage, i.e., how to lessen the scale and improve the nature of bug information. We consolidate occasion choice with highlight choice to all the while lessen information scale on the bug measurement and the word measurement. To decide the request of applying example determination and highlight choice, we remove characteristics from verifiable bug information sets and assemble a prescient model for another bug information set. We exactly inspect the execution output of information lessening on exactly 600,000 bug reports of two huge open source ventures, to be specific Eclipse and Mozilla. Our system work gives an idea to deal with utilizing systems on information handling to shape lessened and excellent bug information in software advancement and support.

B. Efficient Bug Triaging Using Text Mining

Vast open source software ventures get rich rates of submitted bug reports. Triaging these approaching reports physically is blunder inclined and tedious. The objective of bug triaging is to allocate conceivably experienced engineers to new-coming bug reports. To lessen time and cost of bug triaging, we exhibit a programmed way to deal with anticipate an engineer with significant experience to comprehend the new coming report. In this paper, we research the utilization of five term determination strategies on the precision of bug task. Furthermore, we re-adjust the load between engineers taking into account their experience. We direct analyses on four genuine datasets. The exploratory results demonstrate that by selecting a little number of segregating terms, the F-score can be essentially moved forward.

C. Supporting Bug Investigation using History Analysis

In my exploration, I propose a computerized strategy to bolster bug examination by utilizing a novel investigation of the historical backdrop of the source code. Amid the bug-altering process, engineers spend a high measure of manual exertion exploring the bug with a specific end goal to answer a progression of inquiries regarding it. My exploration will bolster engineers in noting the accompanying inquiries concerning a bug: Who is the most suitable designer to settle the bug?, Where is defect found?, When was the bug embedded? Furthermore, Why was the bug embedded?

D. Automated, Highly- Accurate, Bug Assignment Using Machine Learning and Tossing Graphs

The quantity of reported bugs in expansive open source undertakings is high and triaging these bugs is an essential problem in software support. As a stage in the bug triaging process, relegating another bug to the most suitable engineer to alter it, is not just a period devouring and dull errand. The triage, the individual who considers a bug and allocates it to a designer, additionally should know about engineer exercises at various parts of the undertaking. It is clear that just a couple of engineers have this capacity to complete this progression of bug triaging. The primary objective of this paper is to recommend another way to deal with the procedure of performing programmed bug task. The data expected to choose the best designers to settle another bug report is separated from the form control archive of the venture. Not at all like all the past proposed approaches which utilized Machine Learning and Information Retrieval strategies, this examination utilizes the Information Extraction (IE) techniques to remove the data from the product archives. The proposed approach does not utilize the data of the bug storehouse to settle on choices about bugs keeping in mind the end goal to get better results on tasks which don't have numerous altered bugs. The point of this examination is to prescribe the genuine fixers of the bugs. Utilizing this methodology, we accomplished 62%, 43% and 41% correctness on Eclipse, Mozilla and Gnome ventures, individually.

E. Formal Models for Expert Finding in Enterprise Corporation

Hunting an association's report storehouses down specialists gives a practical answer for the assignment of master finding. We introduce two general techniques to master looking given a report accumulation which are formalized utilizing generative probabilistic models. The first of these straightforwardly models a specialist's learning in light of the reports that they are connected with, whilst the second finds archives on theme, and afterward finds the related master. Shaping solid affiliations is urgent to the execution of master discovering frameworks. Subsequently, in our assessment we look at the changed methodologies, investigating an assortment of relationship alongside other operational parameters, (for example, topicality). Utilizing the TREC Enterprise corpora, we demonstrate that the second technique reliably outflanks the first. A correlation against other unsupervised methods, uncovers that our second model conveys superb execution.

III. PROPOSED SYSTEM

A. Bug Triage

Bug triage is a process to assign new bug to appropriate developer. When bug is identified by developer or tester, bug is need to solve in less time. Hence bug must assign to correct developer.

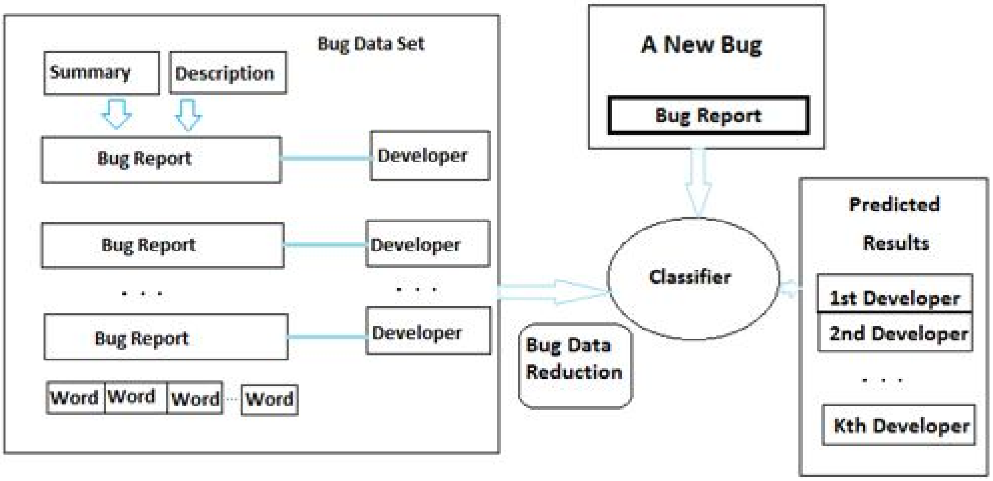

Fig -1: Architecture of proposed system

B. Data Reduction

Reduction of bug data is a process of reducing data for improving bug data quality which combine feature selection and instance selection on sample bug data. This generate reduce bug data, which is high quality and low scale. Due to data reduction bug triage analysis becomes more efficient and appropriate. Column wise and row wise reduction are two ways of data reduction.

Original high scaled Bug Data
Reduced Low scale and high quality Bug Data

Fig -1: Data Reduction.

C. Instance Selection

It is process of selecting instances, which has high quality information. Instance selection process removes non required or low quality instance record. It gives subset of complete dataset, which has same accuracy of original dataset.

D. Feature Selection

An attribute selection method, which is subset of complete attribute of dataset [1]. New attributes gives more efficient data than original attributes. Small representation of bug data solves tasks efficiently.

1) Heuristic methods

a) Step-wise forward selection

b) Step-wise backward elimination

2) Ggeneral methodologies are

a) Feature extraction.

b) Domain-specific.

c) Mapping data to new space (see: data reduction)

E. Naive Bayes Algorithm for Classification

1) TRAINMULTINOMIALNB(C, D)

```
a) V ← EXTRACTVOCABULARY (D)
b) N ← COUNTDOCS(D)
c) for each c ∈ C
d) do $N_c$ ← COUNTDOCSINCLASS( D, c )
e) prior[c] ← $N_c /N$
f) $text_c$ ← CONCATENATETEXTOFALLDOCSINCLASS(D, C )
g) for each  t ∈ V
h) do $T_{ct}$ ← COUNTTOKENSOFTERM(text, t)
i) for each  t ∈ V
j) do condprob[t][c] ← $T_{ct}+1 / \Sigma(T_{ct'}+1)$
k) return V, prior, condprob

2) APPLYMULTINATIONALNB(C,V, prior, condprob )
a) W ← EXTRACTTOKENSFROM(V, d)
b) for each  c ∈ C
c) do score[c] ← log prior[c]
d)   for each t ∈ W
e) do score[c]+= log condprob[t][c]
f) return $argmax_{c∈C}$ score[c]
```

IV. RESULT ANALYSIS

Result Description	*Expected Output*	*Actual Output*
The reduced data must only be used for further analysis. No other attribute must be involved in data processing like k-means clustering and data analysis.	The bug analysis and employee analysis must be shown in graphical manner using pie charts and bar charts for bug basis and individual employee basis.	The proposed system give the efficiency in reducing the ample of data and gives graphical analysis of the developer / tester efficiency based on working time and the work efficiency measure categorized using whether the bug is resolved in first attempt or is reassigned etc.

Table 1. Result Analysis

V. CONCLUSIONS

Proposed system gives reduced bug data using feature selection and instance selection technique. Further analysis is occurred on reduced data to get more efficiency. This system provides graphical analysis of bug data.This system can be useful for analyzing bug data and efficiency of employees. Proposed system minimizes manual effort of bug fixing and minimizes time required for bug fixing as bug triage data is reduced. Now it provides bug data reduction and graphical analysis of previous data. Proposed system also predict appropriate developer to resolve bug using classification .

VI. ACKNOWLEDGMENT

I would like to take this opportunity to express my profound gratitude and deep regard to my guide for his exemplary guidance, valuable feedback and constant encouragement throughout the duration of the project. His valuable suggestions were of huge help throughout my project work. His perceptive criticism kept me working to make this project in a much better way. Working under him was an extremely knowledgeable experience for me.

REFERENCES

[1] 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] Mamdouh Alenezi and Kenneth Magel, Shadi Banitaan "Efficient Bug Triaging Using Text Mining." 2013 academy publisher.

[3] Francisco Servant "Supporting Bug Investigation using History Analysis" 978-1-4799-0215- 6/13 c 2013 IEEE.

[4] Pamela Bhattacharya, Iulian Neamtiu, Christian R. Shelton, "Automated,Highly- Accurate,"Bug Assignment Using Machine Learning and Tossing Graphs." May 2, 2012.

[5] K.Balog, L.Azzopardi, and M. de Rijke, "Formal models for expert finding in enterprise Corpora," in Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inform. Retrieval, Aug. 2006, pp. 43-50.

[6] P. S. Bishnu and V. Bhattacherjee, "Software fault prediction using quad tree-based k-means clustering algorithm," IEEE Trans. Knowl. Data Eng.,vol. 24, no. 6, pp. 1146-1150, Jun.2012.

[7] H.Brighton and C.Mellish, "Advances in instance selection for instance based learning algorithms," Data Mining Knowl. Discovery, vol. 6, no. 2, pp. 153172, Apr. 2002. A. K. Uysal and S. Gunal,"A novel probabilistic feature selection method for text classification," Knowledge-Based Systems, vol. 36, no. 0, pp. 226-235, 2012.