

Towards Bug Triage with Software Data Reduction Technique

Pushpaleela S ME., Sheelagowr P ME., Ass prof,

Abstract— Handling a historical bug triage is a one of the problem while developing a project, have to handle bug triage for reduce the scale and improve the quality of the bug data. We propose an efficient system to reduce high quality bug data to maintain the software development and maintenance

Index Terms—bug triage , data reduction , feature selection,Instance selection.

I. INTRODUCTION

Mining software repositories is an interdisciplinary domain, which is used to employ data mining to deal with software engineering problems. the software development, the large-scale databases for storing the output of software development are called as software repositories. The Datamining has a promising means to handle software data. A bug report, plays an important role in managing software bugs. Software bugs are difficult and software development is

Manuscript received, April 2016.

Pushpaleela S, Computer science, VelsUniversity, Chennai, India,

Sheelagowr p, computer science, VesUniversity, Chennai, India.

expensive in fixing bugs .45 percent of cost are spend in MNCs in bugs fixing . In a bug repository, are formed as a bug and its maintained as a bug report, which records the textual description of reproducing the bug and updates according to the status of bug fixing. bug reports in a bug repository are called bug data.

II. PROCEDURE FOR PAPER SUBMISSION

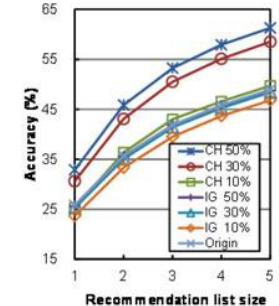
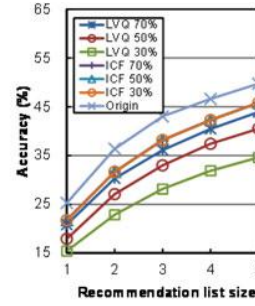
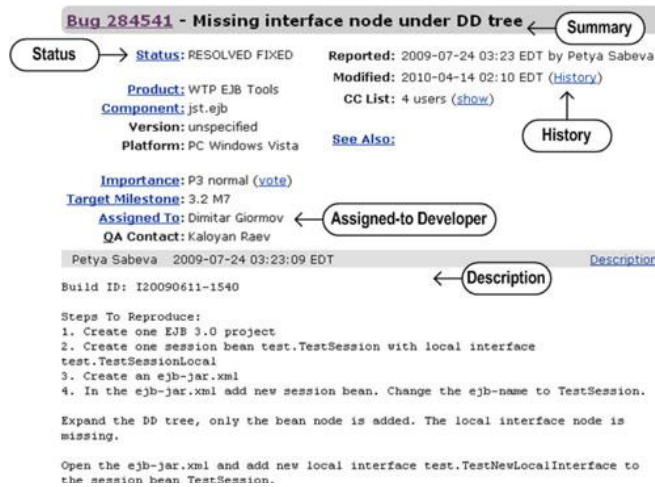
New bugs are manually triaged by an expert by traditional software development, developer, and the lack of expertise of all the bugs, manual bug triage is expensive in time cost and low in accuracy. To avoid the expensive cost of manual bug triage, the work has proposed an automatic bug triage approach, to reduce the bug *data text classification techniques are used*

A. BUG TRIAGE:

Bug repositories are mainly used for maintaining software Bugs, when a software bug is found; a reporter sendes his bug to the bug repository. A recorded bug are called a bug report, the two key items about the information of the bug are summary and description , that are recorded in ordinary languages. The summary denotes a general statement for identifying a bug while the description

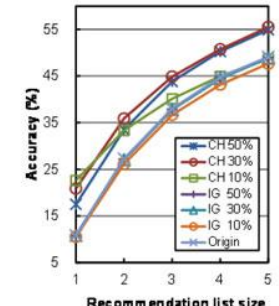
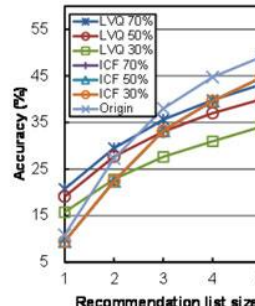
gives the details for reproducing the bug. Some other items are stored in a bug report for future use for bug, such as the product, the platform, and the importance. Once a bug report is formed, a human trigger assigns this bug to a developer

B. Figures



(a) Instance selection in Eclipse

(b) Feature selection in Eclipse



(c) Instance selection in Mozilla

(d) Feature selection in Mozilla

Applying Instance Selection and Feature Selection

Data processing techniques are used in both instance and feature selection. For a given data set in a certain application, instance selection are used as a subset of relevant instances while the feature selection aims to update a subset of relevant features, we use to made a combination of instance selection and feature selection.



Pre-Processor:

We aim to augment the data set to build a preprocessing approach, that are applied before an existing bug triage approach. In contrast to modeling the textual content of bug reports in existing work

Predication for reduction order

An instance selection algorithm IS and a feature selection algorithm FS, FS to IS and IS to FS are viewed as two orders for applying reducing techniques. Hence, a main thing is how to determine the order of reduction techniques, i.e., how to choose one between FS to IS and IS to FS. Refer to this problem as the prediction for reduction orders

III REDUCTION ORDER

To apply the data reduction for each new bug data set, need to check the accuracy of both two orders (An instance selection algorithm IS and a feature selection algorithm FS, FS to IS and IS to FS) and need to choose a better one. To avoid the time cost of manually checking in both reduction orders, we consider predicting the reduction order for a new bug data set based on historical data sets.

Login Module:

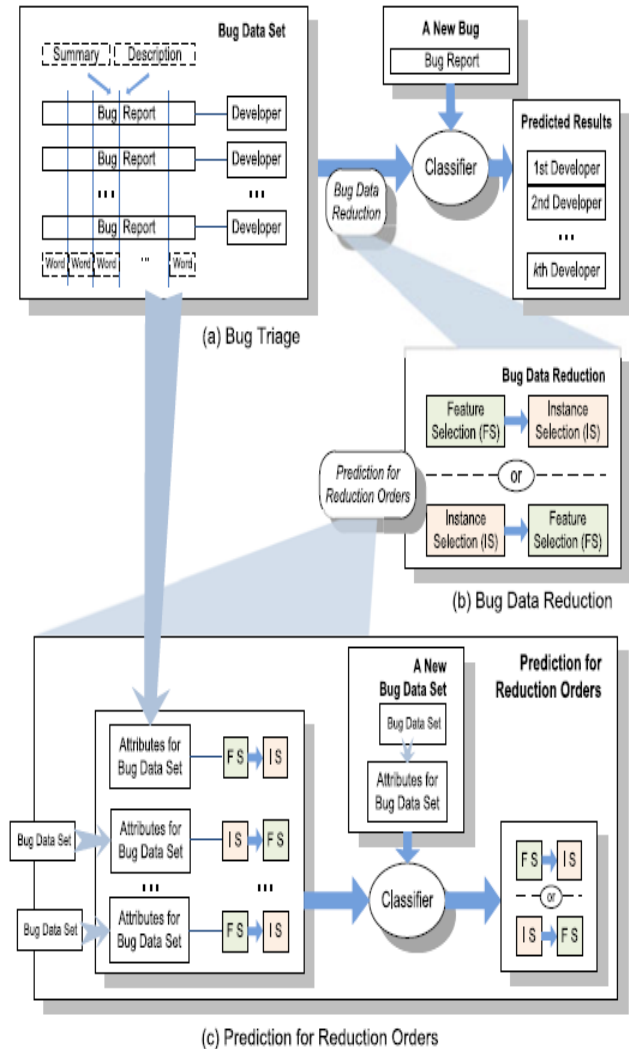
In login module there are two types of login, Employee login, Admin login, to authenticate this process Username

and Password are used and MySQL database is used for store the data. The both username and password are stored in related database. During the login process the given username and password are verified from the database. Only Valid user can access the information and other services. In the verification process the give username and password are incorrect we cannot access the information.

Data Reduction:

Here data reduction is used, to save the labor cost of developers, the data reduction for bug triage has two goals.

1. The data scale to be Reduced
2. Improving the accuracy of bug triage



HELPFUL HINTS

A. Reference

An Overview of Attributes for a Bug Data Set

Index Attribute name Description

- B1 # Bug reports Total number of bug reports.
- B2 # Words Total number of words in all the bug reports.
- B3 Length of bug reports Average number of words of all the bug reports.
- B4 # Unique words Average number of unique words in each bug report.
- B5 Ratio of sparseness Ratio of sparse terms in the text matrix. A sparse term refers to a word with zero frequency in the text matrix.
- B6 Entropy of severities Entropy of severities in bug reports. Severity denotes the importance of bug reports.
- B7 Entropy of priorities Entropy of priorities in bug reports. Priority denotes the level of bug reports.
- B8 Entropy of products Entropy of products in bug reports. Product denotes the sub-project.
- B9 Entropy of components Entropy of components in bug reports. Component denotes the sub-sub-project.
- B10 Entropy of words Entropy of words in bug reports.
- D1 # Fixers Total number of developers who will fix bugs.
- D2 # Bug reports per fixer Average number of bug reports for each fixer.
- D3 # Words per fixer Average number of words for each fixer.

B. Algorithm

Data reduction based on FS!IS

Input: training set T with n words and m bug reports, reduction order FS!IS

final number nF of words,

final number mI of bug reports,

Output: reduced data set T FI for bug triage

- 1) apply FS to n words of T and calculate objective values for all the words;
- 2) select the top nF words of T and generate a training set T F ;
- 3) apply IS to mI bug reports of T F ;
- 4) terminate IS when the number of bug reports is equal to or less than mI and generate the final training set T FI .

C. Equations

Accuracy $k \frac{1}{4}$ = correctly assigned bug reports in k candidates

all bug reports in the test set .

Loss $k \frac{1}{4}$ = Accuracy k by origin_ Accuracy k by ICF

Accuracy k by origin

Loss $k \frac{1}{4}$ = Accuracyk by origin_Accuracyk by ICF

Accuracyk by origin ,

IV EDITORIAL POLICY

The submitting author is responsible for obtaining agreement of all coauthors and any consent required from sponsors before submitting a paper. It is the obligation of the authors to cite relevant prior work.

Authors of rejected papers may revise and resubmit them to the journal again.

V PUBLICATION PRINCIPLES

The contents of the journal are peer-reviewed and archival. The journal INTERNATIONAL JOURNAL OF SCIENCE, ENGINEERING AND TECHNOLOGY RESEARCH (IJSETR) publishes scholarly articles of archival value as well as tutorial expositions and critical reviews of classical subjects and topics of current interest.

Authors should consider the following points:

- 1) Technical papers submitted for publication must advance the state of knowledge and must cite relevant prior work.
- 2) The length of a submitted paper should be commensurate with the importance, or appropriate to the complexity, of the work. For example, an obvious extension of previously published work might not be appropriate for publication or might be adequately treated in just a few pages.
- 3) Authors must convince both peer reviewers and the editors of the scientific and technical merit of a paper; the standards of proof are higher when extraordinary or unexpected results are reported.
- 4) Because replication is required for scientific progress, papers submitted for publication must provide sufficient information to allow readers to perform similar experiments or calculations and use the reported results. Although not everything need be disclosed, a paper must contain new, useable, and fully described information. For example, a specimen's chemical composition need not be reported if the main purpose of a paper is to introduce a new measurement technique. Authors should expect to be challenged by reviewers if the results are not supported by adequate data and critical details.

III. CONCLUSION

To minimize the labor cost and time cost bug triage is used here to reduce the scale of bug data sets as well as improve the data quality by combining feature selection with instance selection. To define the order of applying instance selection and feature selection for a new bug data set, we extract attributes of each bug data set and train a predictive model based on historical data sets. In future work, we plan to mainly improve the results of data reduction in bug triage to explore how to prepare a high quality bug data set and tackle a domain specific software task. .

Pushpaleela s received the BE degree in computer science in 2014 and currently a student of vels university by doing the ME in computer science of engineering

Sheelagowr p received the BE degree in computer science and ME degree in same field and currently doing the PhD degree

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable and constructive comments on improving this paper. This work was supported by the INTERNATIONAL JOURNAL OF SCIENCE, ENGINEERING AND TECHNOLOGY RESEARCH (IJSETR)

REFERENCES

- [1] Jifengxuan, HeJiang, Member, IEEE, YanHu, ZhileiRen, WeiqinZou, ZhongxuanLou, and XindongWu, Fellow, IEEE, "Towards effective bug triage with software data reduction techniques" Jan, pp. 1041-4347
- [2] J. Anvik, L. Hiew, and G. C. Murphy, "Who should fix this bug?" in Proc. 28th Int. Conf. Softw. Eng., May 2006, pp. 361–370.
- [3] S. Artzi, A. Kiezun, J. Dolby, F. Tip, D. Dig, A. Paradkar, and M. D Ernst, "Finding bugs in web applications using dynamic test generation and explicit-state model checking," IEEE Softw., vol. 36, no. 4, pp. 474–494, Jul./Aug. 2010.
- [4] 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.