International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 11 | Nov -2016 www.irjet.net p-ISSN: 2395-0072

Case study on

Software Data Reduction Techniques use for Effective Bug Triage

Vishal Patil, Rahul Patil, Siddheshwar Pawar,Sandip Patil

Vishal Patil , SKNSITS LONAVALA Tq:-maval Dist:-Pune Pin:-410401

Siddheshwar Pawar , SKNSITS LONAVALA Tq:-maval Dist:-Pune Pin:-410401

Guide: Prof.Neha Jamdar, Dept. of Computer Engineering, SKNSITS college lonavala, Maharashtra,India

---

Abstract - *A bug repository an important part in overseeing software bugs. Many open source programming ventures have an open bug repository that permits both developers and users to submit imperfections or problem in the product, recommend conceivable upgrades, and remark on existing bug reports. For open source large-scale software projects, the number of daily bugs is so large which makes the triaging process very hard and challenging. Software organizations spend more than 45 percent of cost in fixing bugs .There are two difficulties identified with bug information that may influence the effective use of bug repositories in software development tasks, namely the large scale and the low quality. In a bug repository, a bug is kept up as a bug report, which records the printed description of imitating the bug and upgrades as indicated by the status of bug fixing.*

Key Words: **Software companies ,Software bugs ,Data reduction ,Developer ,Manager**.

1.INTRODUCTION

MINING SOFTWARE REPOSITORIES IS AN INTERDISCIPLINARY DOMAIN, WHICH EXPECTS TO UTILIZE DATA MINING TO MANAGE SOFTWARE ENGINEERING PROBLEMS. IN ADVANCED PROGRAMMING IMPROVEMENT, SOFTWARE REPOSITORIES ARE LARGE-SCALE DATABASES FOR STORING THE OUTPUT OF SOFTWARE DEVELOPMENT, E.G., SOURCE CODE, BUGS, EMAILS, AND SPECIFICATIONS. BY UTILIZING DATA MINING PROCEDURES, MINING SOFTWARE REPOSITORIES CAN REVEAL INTERESTING DATA IN SOFTWARE REPOSITORIES AND SOLVE REAL WORLD SOFTWARE PROBLEMS .A BUG REPOSITORY (A TYPICAL SOFTWARE REPOSITORY, FOR STORING DETAILS OF BUGS), ASSUMES AN ESSENTIAL PART IN MANAGING SOFTWARE BUGS .SOFTWARE BUGS ARE UNAVOIDABLE AND FIXING BUGS IS COSTLY IN SOFTWARE DEVELOPMENT.

1.1 PROPOSED SYSTEM

- In this part, we display the information arrangement for applying the bug data reduction. Introduce the issue of data reduction for bug triage.
- This issue plans to expand the data set of bug triage in two viewpoints, namely
  a) Reduce the sizes of the bug measurement and the word measurement.
  b) To enhance the accuracy of bug triage.
- Propose a combination approach to deal with tending to the issue of data reduction. This can be viewed as an application of instance selection and feature selection in bug repositories.
- Build a binary classifier anticipate the request of applying instance selection and feature selection.

1.2 Algorithm

Data reduction based on FS→IS

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

final number n_F of words,

final number m_I 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 n_F words of T and generate a training set T_F;

3) apply IS to m_I bug reports of T_F ;

4) terminate IS when the number of bug reports is equal to or less than m_I and generate the final training set T_{FI} .

2. System Architecture

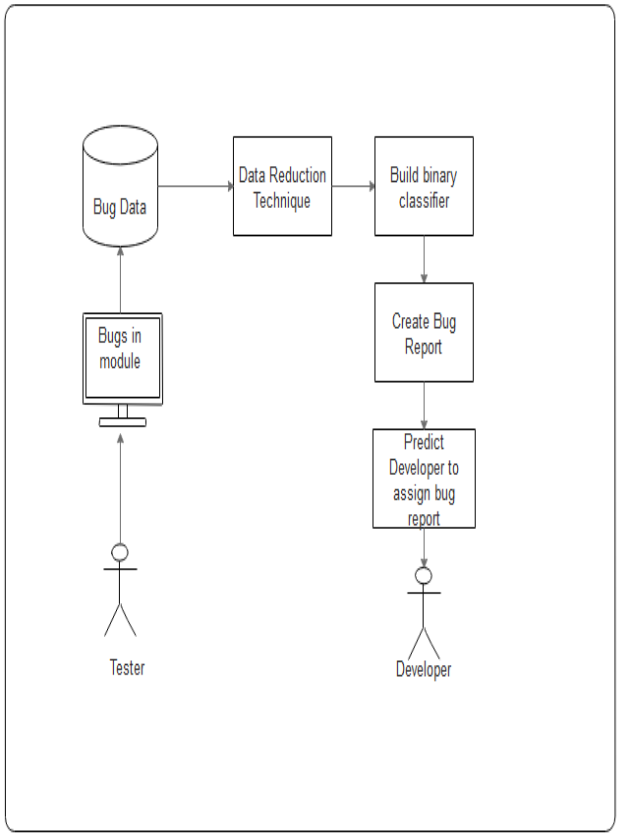

Fig1. System architecture.

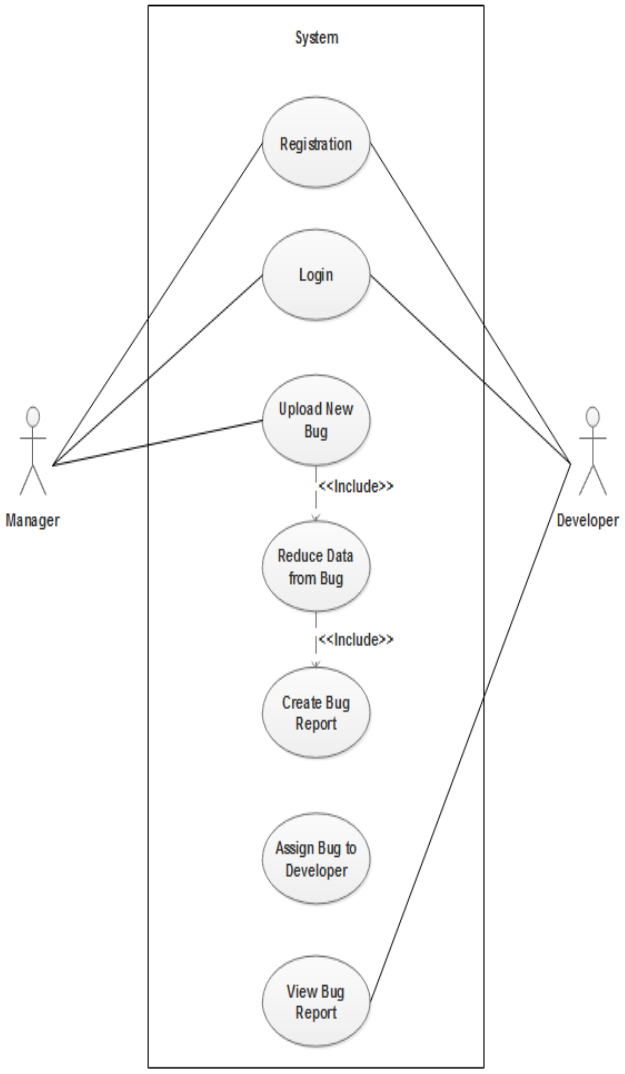

Fig2. Use Case Diagram.

3. RESULTS AND DISCUSSION

In this paper, we propose the problem of data reduction for bug triage to diminish the size of data sets and to enhance the quality of bug reports. We use method of instance selection and feature selection to reduce noise and redundancy in bug data sets. However, not all the noise and redundancy are removed. For example only less than 50 percent of duplicate bug reports can be removed in data reduction (198=532 ¼ 37:2% by CH ! ICF and 262=532 ¼ 49:2% by ICF ! CH). The reason for this fact is that it is difficult to correctly detect noise and redundancy in real-world applications. On one hand, due to the big size of bug repositories, there exist no adequate labels to mark whether a bug report or a word belongs to noise or redundancy; on the other hand, since all the bug reports in a bug repository are recorded in natural languages, even noisy and redundant data may contain useful data for bug fixing.

4. Conclusion

Bug triage is an costly step of software maintenance in both labour cost and time cost. In this paper, we mix feature selection with instance selection to reduce the size of bug data sets as well as enhance the data quality. To determine 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. We empirically investigate the data reduction for bug triage in bug repositories of two large open source projects, namely Eclipse and Mozilla. Our work provides an approach to leveraging techniques on data processing to form reduced and high-quality bug data in software development and maintenance. In future work, we plan on improving 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. For predicting reduction orders, we

plan to pay efforts to find out the potential relationship between the attributes of bug data sets and the reduction orders.

.

REFERENCES

1. [1] 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.

2. [2] S. Artzi, A. Kie_zun, 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.

3. [3] 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.

4. [4] C. C. Aggarwal and P. Zhao, "Towards graphical models for text processing," Knowl. Inform. Syst., vol. 36, no. 1, pp. 1–21, 2013.

5. [5] Bugzilla, (2014). [Online]. Avaialble: http://bugzilla.org/

6. [6] 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.

7. [7] 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.

8. [8] H. Brighton and C. Mellish, "Advances in instance selection for instance-based learning algorithms," Data Mining Knowl. Discovery, vol. 6, no. 2, pp. 153–172, Apr. 2002.

9. [9] 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, pp. 301–310.

10. [10] V. Bol_on-Canedo, N. S_anchez-Maro˜no, and A. Alonso-Betanzos, "A review of feature selection methods on synthetic data," Knowl. Inform. Syst., vol. 34, no. 3, pp. 483–519, 2013.