Using Data Reduction Techniques for Effective Bug Triage

Shanthipriya. D[1], Deepa.K[2]
[1] Department of Computer Science & Engineering
[2] Department of Information Technology
[1, 2] Sri Ramakrishna Engineering College,Coimbatore – 641 022, TamilNadu, India

Abstract- *An automatic bug triage process is an inevitable step to fix the software bugs. To decrease the manual and time cost, text classification techniques are applied to perform the automatic bug triage. The main goal of essential bug triaging software is to allocate possibly experience developers to new coming bug reports. The existing bug triage approach suffers from large scale and low quality bug data. The proposed system employs the combination of feature selection algorithm (FS) and instance selection algorithm (IS) for bug triage. These data reduction techniques are used to shrink the bug data and also to enhance the accuracy. The performance of proposed system is evaluated by using Mozilla bug data set. To show the effectiveness, scales of bug data is reduced to avoid the manual and time cost, upgrades the accuracy of bug triage with standard bug data in software maintenance.*

Keywords- Bug Triage, Feature selection, Instance selection, Mining Software repository.

I. INTRODUCTION

A Software bug is an issue causing a program to collapse or create unacceptable output. The problem is caused by inadequate or invalid logic. A bug can be an error, mistake, flaw or fault, which may cause collapse or variation from usual results. Most bugs are due to human errors in source code or its design. A program is said to be buggy when it includes a huge number of bugs, which concern program functionality and cause erroneous results [5]. The details of bugs are stored in large database which is named as bug repository or bug tracking system [2]. An open source bug repository [2], which is employed by many large software companies for open source projects i.e., Mozilla [9]. To solve the real world engineering issues some data mining methods [16] are exercised to describe with some useful information accumulated in bug ordnance. Bug Triage is the process to assign relevant developer to each bug reports in order to fix it [19].

Due to huge number of daily bugs and lack of skill person of all the bugs, manual triage is an expensive in time cost and labor cost, low in precision. To defeat the limitations of existing work, an automatic bug triage approach is proposed [17]. This approach applies the text classification techniques in order to expect the valid developer for bug reports without tossing.

In this paper, we proposed the data reduction techniques using the combination of the instance selection algorithm (IS) [4] and feature selection algorithm (FS) [11, 12] . These approaches are used to shrink the data scale and also improve the accuracy of bug data set. The order of applying the reduction techniques may concern the consequence of bug triage approach. In this paper, to determine the order of bug data reduction techniques, i.e., FS to IS or IS to FS we propose a Predictive model [17]. The text classification technique i.e., Naive Bayes is used to predict correct developer to solve and fix the bug reports [21] in order to shrink the manual triager cost. The proposed system performance is verified using Mozilla bug data set [9] which obtains 78% accuracy after the training set reduction. The outcome shows that the experiment on reduce training sets can obtain better accuracy than that on original training set.

The remainder section of this paper is organized as follows: Section 2 explains the proposed methodology. Section 3 explains the experimental results and discussion. Section 4 lists the related work. In Section 5 we briefly conclude this paper and present our future work.

II. RELATED WORKS

As our Knowledge, there is no combination of data reduction methods in turn to decrease the data scale and upgrade the exactness of bug triage approach in the illustration.

Jeong, Kim, Zimmermann introduced a tossing graph model based on Markov property from the conception of reassign the bug reports to other developers [6]. Shivaji and colleagues [12] proposed the feature selection techniques to predict the software bugs. Anvik, L. Hiew, and G. C. Murphy [1] extend the machine learning approaches. They describe the bug triage as semi-supervised approach which updated with weighted recommendation list; based on the probabilistic view the relevant developers are employed to the human triage [4,

13]. Cubranic and Murphy [3] projected supervised learning method (NB Classifier) to assist in bug triage by using text categorization to predict the relevant developers. A classification model should be designed to investigate the relationship among the datas in bug data set and to check the quality [20, 17].

Fu.Y, Zhu.X, and Li.B [4] investigated to obtain the accurate prediction model with minimum cost by labelling most informative instances. In contrast to these papers, our paper aims to employ the information gain algorithm to develop the software value of bug data prediction. In this paper, we focus on the issue of bug data reduction and low in precision of bug data set. Further the combination of feature selection and instance selection algorithm intend to shrink the bug data set and develop the performance of bug triage with high-quality bug data in software maintenance and improvement.

III. NEED FOR BUG TRIAGE PROCESS

Bug triage is an important process in bug fixing process in order to assign relevant developers to new coming bugs. Fig.1 represents the bug triage process [19]. Some of the steps involved in bug triage process are

1. Find bugs to triage
2. Pre-filter bug reports
3. Search for duplicates of bugs
4. Check information provided in bug report
5. Attempt to reproduce bug
6. Set bug status
7. Prioritize bug

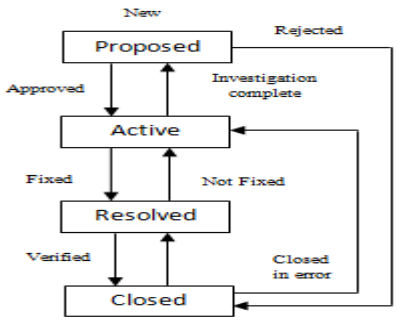

Fig.1 Bug Triage Process

8. Notify developers – needed only in very specific cases if bug seems to be a blocker / critical.

IV. PROPOSED METHODOLOGY

The fig.2 illustrated the system architecture of the proposed system. Bug datas of Mozilla are taken from an open source bug repository i.e., Bugzilla. This open source bug repository contains all information about the software bugs. Each bug has the bug statement and the details of the developer who employed on that particular bug. The bug details may be divided into two parts: summary and description. The proposed system can use bug data reduction technique which reduces labor cost and time cost. Here, the bug data reduction method is used to prepare the content for bug triage. This proposed system mainly concerns on two goals. First, reduces the data scale and second, improves the accuracy of bug data.

The instance selection and feature selection are pre-processing techniques which are used for bug data reduction.

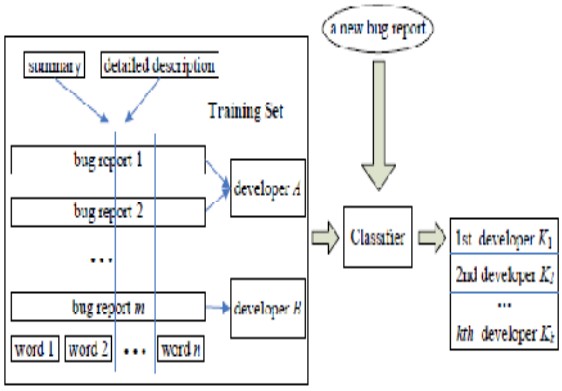

Fig.2 the Text Categorization approach for Bug triage

For a specified bug data set, the instance selection is applied to find the significant subsets (i.e., bug reports in bug data set) and after/ before feature selection is applied to find the subset of appropriate features (i.e., words in bug data set). In proposed system, the combination of these techniques is used.

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,
  1. apply FS → n words of T
  2. calculate objective values for all the words
  3. select the top n_F words of T
  4. generate a training set T_F
  5. apply IS →m_I bug reports of T_F
  6. terminate IS when the number of bug reports is equal to or less than m_I
  7. Generate the final training set T_{FI}.

Output:

- reduced data set T_{FI} for bug triage

By applying these techniques, the bug data scale can get reduced and also upgrades the performance of the bug triage approach. The predictive model is proposed in order to predict the correct order to shrink the bug data set. By employing this model FS to IS or IS to FS order can be predicted without any complication. The text classification approach i.e., Naive Bayes is used to predict the correct developer for the predicted bug. C4.5 AdaBoost is used to calculate the precision, recall and to balance this F measure values are calculated. The accuracy of Mozilla bug data set can be calculated as 78% which reduces data scale and improves the performance of bug triage approach.

V. RESULTS AND DISCUSSION

The performance of bug data set can be measured by using both training and test bug data set. In this attributes of each training and test bug data set can be calculated. The attributes are named as bug dataset details as B1 to B10 and developer details as D1 to D8. The pre-processing techniques for data reduction i.e., feature selection and instance selection is applied to in training bug data set.

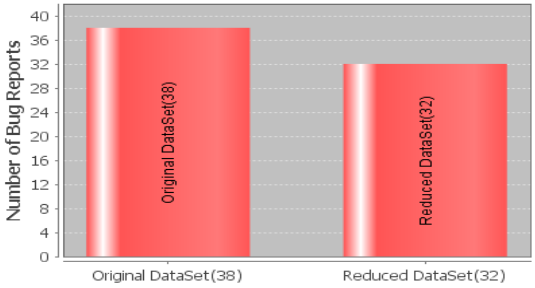

Fig.3 Comparison result between original and reduced bug data set (Training bug data Set)

The training data set contains 40 bug records which give complete information about the bug data stored in large database i.e., Bugzilla.Fig.3 illustrates the comparison between original bug data set and reduced bug data set.

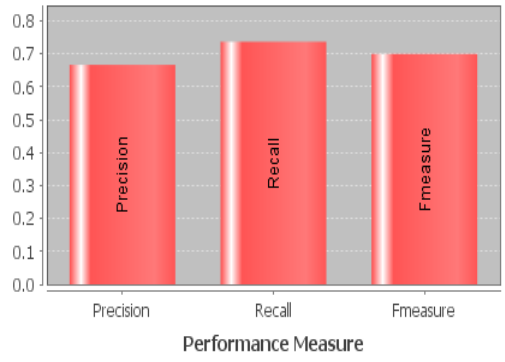

Fig.4 Comparison Graph for Precision, Recall and F- measure

The classifier i.e., Naive Bayes is trained by training data set with their data reduction order. Then, the classifier is used to predict the correct order to test data set and reduce the labor cost. By this, the bug triage approach is upgraded by their performance. Fig.4 illustrates the precision, recall, and F-measure values of Mozilla bug data set are 0.667, 0.737 and 0.70. The accuracy is measured as 78% by using Naive Bayes classifier for training data set.

VI. CONCLUSION

Bug triage is an important and significant step of software protection in both labor cost and time cost. The proposed method combines the feature selection algorithm (FS) with instance selection algorithm (IS) in order to trim down the scale of bug data sets as well as develop the data value. A Predictive model is utilized to establish the order of applying reduction order, i.e., FS to IS or IS to FS. The proposed system performance is verified using Mozilla bug data set. To exhibit the value, a scale of data set is condensed by using data reduction technique in order to diminish the time and labor cost, upgrades the precision of bug triage with high-quality bug data in software progress and maintenance.

The future work of the proposed system is to get better the outcome of data reduction in bug triage to investigate how to organize a high quality bug data set and deal with a domain-specific software assignment. For predicting reduction orders, aim to give attempts to locate out the possible relationship among the attributes of bug data sets and the reduction orders.

REFERENCES

[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] Bugzilla, (2015). [Online]. Avaialble: http://bugzilla.org/

[3] D.Cubranic and G. C. Murphy, “Automatic bug triage using text categorization,” in Proc. 16th Int. Conf. Softw. Eng. Knowl. Eng., Jun. 2004, pp. 92–97.

[4] Y. Fu, X. Zhu, and B. Li, “A survey on instance selection for active learning,” Knowl. Inform. Syst., vol. 35, no. 2, pp. 249–283, 2013.

[5] https://www.techopedia.com/definition/24864/software-bug

[6] G. Jeong, S. Kim, and T. Zimmermann, “Improving bug triage with tossing graphs,” in Proc. Joint Meeting 12th

Eur. Softw. Eng. Conf. 17th ACM SIGSOFT Symp. Found. Softw. Eng., Aug. 2009, pp. 111–120.

[7] S. Kim, H. Zhang, R. Wu, and L. Gong, "Dealing with noise in defect prediction," in Proc. 32nd ACM/IEEE Int. Conf. Softw. Eng., May 2010, pp. 481–490.

[8] D. Matter, A. Kuhn, and O. Nierstrasz, "Assigning bug reports using a vocabulary-based expertise model of developers," in Proc. 6th Int. Working Conf. Mining Softw. Repositories, May 2009, pp. 131–140.

[9] Mozilla. (2015). [Online]. Available: http://mozilla.org/

[10] 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.

[11] M. Rogati and Y. Yang, "High-performing feature selection for text classification," in Proc. 11th Int. Conf. Inform. Knowl. Manag., Nov. 2002, pp. 659–661.

[12] S. Shivaji, E. J. Whitehead, Jr., R. Akella, and S. Kim, "Reducing features to improve code change based bug prediction," IEEE Trans. Soft. Eng., vol. 39, no. 4, pp. 552–569, Apr. 2013.

[13] 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, pp. 253–262.

[14] X. Wang, L. Zhang, T. Xie, J. Anvik, and J. Sun, "An approach to detecting duplicate bug reports using natural language and execution information," in Proc. 30th Int. Conf. Softw. Eng., May 2008, pp. 461–470.

[15] D. R. Wilson and T. R. Mart_ınez, "Reduction techniques for instance-based learning algorithms," Mach. Learn., vol. 38, pp. 257–286, 2000.

[16] T. Xie, S. Thummalapenta, D. Lo, and C. Liu, "Data mining for software engineering," Comput., vol. 42, no. 8, pp. 55–62, Aug. 2009.

[17] J. Xuan, H. Jiang, Y. Hu, Z. Ren, Z. Luo, W.Zou and X. Wu, "Towards Effective Bug Triage with Software Data Reduction Techniques" in IEEE Trans. on Knowl. and Data Eng., vol. 27, no. 1, Jan. 2015.

[18] J. Xuan, H. Jiang, Z. Ren, and W. Zou, "Developer prioritization in bug repositories," in Proc. 34th Int. Conf. Softw. Eng., 2012, pp. 25– 35.

[19] J. Xuan, H. Jiang, Z. Ren, J. Yan, and Z. Luo, "Automatic bug triage using semi-supervised text classification," in Proc. 22nd Int. Conf. Softw. Eng. Knowl. Eng., Jul. 2010, pp. 209–214.

[20] T. Zimmermann, N. Nagappan, P. J. Guo, and B. Murphy, "Characterizing and predicting which bugs get reopened," in Proc. 34th Int. Conf. Softw. Eng., Jun. 2012, pp. 1074–1083.

[21] W. Zou, Y. Hu, J. Xuan, and H. Jiang, "Towards training set reduction for bug triage," in Proc. 35th Annu. IEEE Int. Comput. Soft. Appl. Conf., Jul. 2011, pp. 576–581.