ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

A Technique to Reduce Bug Data for Effective Bug Triage

Suvarna Kale, Ajay Kumar Gupta

ME Student, Department of Computer Engineering, SP'S IOK College of engineering Shirur, Savitribai Phule Pune University and Maharashtra India

Assistant Professor, Department of Computer Engineering, SP'S IOK College of engineering Shirur, Savitribai Phule Pune University and Maharashtra India

ABSTRACT: An unavoidable step of handling software bugs is bug triage, which aims to correctly assign a developer to a new bug. Software organizations spend more than 45 percent of cost in handling Software bugs. In order to reduce the time cost in manual work of bug fixing, text classification techniques are used for automatic bug triage. In this work, we address the problem of data reduction for bug triage, i.e., how to lessen the scale and improve the quality of bug data. We combine instance selection with feature selection to simultaneously decrease data scale on the bug dimension and the word dimension. To determine the order of applying instance selection and feature the previously fixed bug reference is used and a predictive model is formed for new bug data set. The results demonstrate that the data reduction can effectively reduce the data scale and improve the accuracy of bug triage.

KEYWORDS: bug triage; mining software repositories; bug data reduction; feature selection; instance selection; text classification

I. INTRODUCTION

In software development process a bug repository plays a vital role in managing software bugs. Software bugs are unavoidable and fixing bugs is costly in software development and maintenance process. Software companies spend more than 45 percent of cost in fixing bugs. In a bug repository, a bug is kept up as a bug report, textual description of reproducing the bug and updates according to the status of bug fixing. A bug repository gives a data platform to support many types of tasks on bugs. In this paper, bug reports are referred to as bug data in a bug repository. In bug repository large number of bug reports is stored.

Due to large number of bug reports there are two difficulties identified with bug data that might affect the effective use of bug repositories in software development process that are large scale and low quality. On one hand, because of the day by day reported bugs, a very large number of new bugs are stored in bug repositories. Due to this large number of daily bugs the manual bug triage is costly in both time and cost. It is a challenging job to manually examine such huge amount of bug data in software development also software techniques suffer from the low quality of bug data. Two specific characteristics of low quality bugs are noise and redundancy.

A. *Background*

Bug repositories are generally used for maintaining software bugs. Once a software bug is found, a reporter (regularly a developer, a tester, or an end user) records this bug to the bug repository. A recorded bug is known as a bug report, which has multiple items for detailing the information of reproducing the bug. In a bug report summary and description are two key items about information of the bug. The summary denotes the general statement for identifying a bug and description gives the details for reproducing the bug. Once a bug report is formed it is assigned to the developer who will try to fix this bug. The developer starts to fix the bug based on historical bug fixing. The developer takes the reference of previous fixed bug e.g. searching for similar bugs and applying existing solutions to new bug. Item history shows the changes in bug report. Manual bug triage takes more time and also is not error free as the numbers of daily reported bugs are huge in number.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

B. *Motivation*

Real world data always consist of noise and redundancy. Noisy data may mislead the data analysis techniques while redundant data increase the cost of data processing. In bug repositories all the bug reports are filled by developer in their own languages. In bug repositories the low quality bugs accumulate with the growth in scale. Such low quality and large scale data may affect the performance of bug fixing. So there is need to reduce the scale and improve the quality of bug data.

C. *Problem Statement*

A time consuming step of handling software bug is bug triage which correctly assign a developer to a new coming bug in software development and maintenance process. In order to decrease the time cost of manual bug triage text classification techniques are used to conduct automatic bug triage with more accuracy. Instance selection and feature selection techniques are used for data reduction.

II. LITERATURE SURVEY

The following table shows the overview on literature survey along with the limitation of each research work.

TABLE I: Overview of Literature Survey

Sr. No.	Paper Name	Author Name	Description	Limitation
1	**Advances In Instance Selection For Instance-based learning Algorithm**	Henry Brighton, Chris Mellish	Performs comparison of the ICF algorithm with RT3 over 30 domains. Argue that neither of these two algorithms is superior both record the highest accuracy and space reduction on certain problems.	In the context of noise removal problem specific dependency is a problem.
2	**What Makes a Good Bug Report?**	Thomas Zimmermann, Rahul Premraj, Nicolas Bettenburg, Sascha Just, Cathrin Weiss.	The most severe problems encountered by developers are errors in steps to reproduce ,incomplete information and wrong observed behavior.	There is mismatch between what information developers consider as important and what users provide.
3	**Towards Training Set Reduction For Bug triage**	Weiqin Zou ,Yan Hu, Jifeng Xuan, He Jiang	It was the first work of combining feature selection with instance selection to reduce the training set for the bug triage problem.70% of words and 50% of bug reports are removed in this approach.	The precision rate is very low

III. RELATED WORK

John Anvik, Lyndon Hiew and Gail C. Murphy had proposed a semi-automated approach of bug assignment. They used a supervised machine learning algorithm that was applied to information in the bug repository. Here the precision level for eclipse was 50% and 64% for firefox. Davor Cubranic and Gail C. Murphy had presented an application of supervised machine learning using a Naïve Bayes classifier to automatically assign bug report to developers. In their approach new bug reports would be automatically assigned to the developer predicted to be the most appropriate to the content. In “predicting the severity of a reported bug” Ahmed Lamkanfi, Serge Demeyer, Giger and Barth Goethals showed how soon a reported bug needs to be fixed partly depends on its severity. They showed that it is possible to predict the severity based on other information contained in the bug report.

Henry Brighton and Chris Mellish performed comparison of the ICF algorithm with RT3 over 30 domains. In their approach they argued that neither of the two algorithms is superior, both record the highest accuracy and space

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

reduction. Weiqin Zou,Yan Hu, Jifeng Xuan, He Jiang had proposed training set reduction i.e. bug data reduction was the first work of combining feature selection with instance selection to reduce the training set i.e. data set for the bug triage problem. 70% of words and 50% of bug reports are removed in their approach.

IV. PROPOSED BUG DATA REDUCTION ALGORITHM

INPUT:

Training set T with n word and m bug reports,
Reduction order IS --> FS
Final number ml of report
Final number nFof word

OUTPUT:

Reduce data set TFI for bug triage
1. Apply IS to ml bug report of TF;
2. Apply FS to n words of T;
3. Select the top nFwords of T and generate a training set TF;
4. Terminate IS when the number of bug report is equal to or less than ml and generate the final training set TFI ;

V. SYSTEM ARCHITECTURE AND MATHEMATICAL MODEL

A. *Architecture*

In proposed architecture we are using ICF algorithm for instance selection, also we are using Greedy algorithm for feature selection and SVM algorithm for text classification.Once the new bug is inserted, at first instance and feature selection is done then SVM is applied.

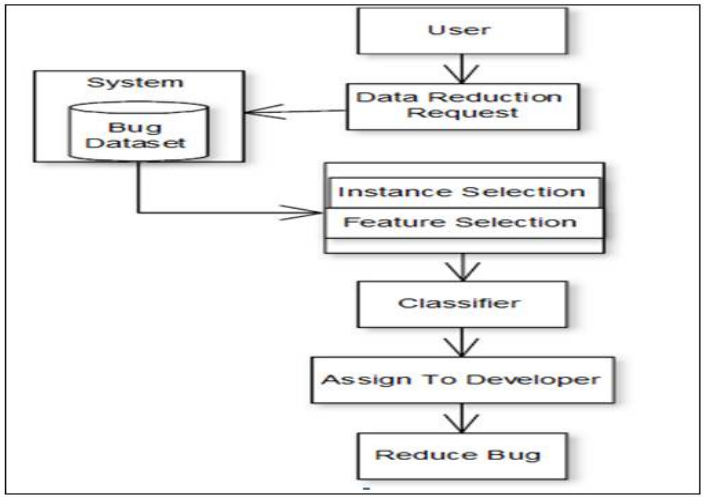

Fig. 1 Proposed System Architecture

B. *Mathematical Model*

The proposed system is represented using mathematical model as follows.

INPUT:-

LET S IS THE WHOLE SYSTEM CONSIST OF

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

S= {U, D, FS, IS, TC, O}
WHERE,
U=USER.
D= {B1, B2… BN}
IS = INSTANCE SELECTION.
FS = FEATURE SELECTION.
TC = TEXT CLASSIFICATION.

PROCEDURE:

INSTANCE AND FEATURE SELECTION
IS → FS = BUG DATA REDUCTION. WHICH FIRST APPLIES IS AND THEN FS.
GIVEN A BUG DATA SET, THE OUTPUT OF BUG DATA REDUCTION IS A NEW AND REDUCED DATA SET.

OUTPUT:

GETTING THE APPROPRIATE DECISION FROM THE ABOVE TECHNIQUE

VI. SYSTEM PERFORMANCE

A. *Accuracy*

The accuracy is a standard method used to measure the performance.The accuracy can be calculated as,

$$accuracy_n = \frac{\#\text{ of correct relevant developers}}{\#\text{ of bug reports in test set}}, \quad n \geq 1$$

Where, n is the size of the recommendation list.

B. *Efficiency*

The efficiency can be calculated by using the hit ratio i.e. the numbers of bugs are properly assigned to the correct potential developer and he resolved it successfully. As shown in figure for top1 developer the hot ratio is 74% and for top 30 it is 98%.

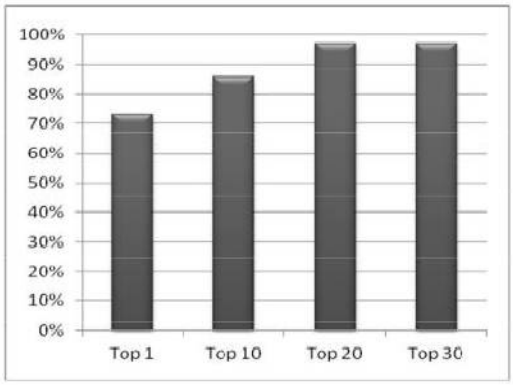

Fig. 2 The expected result on overall efficiency of proposed approach

VII. RESULT OF PRACTICAL WORK

This section shows the practical result of system implementation. In home page all the menu are present. Here user first do the registration then login to the system with the help of his user name and password as shown in figure 3.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

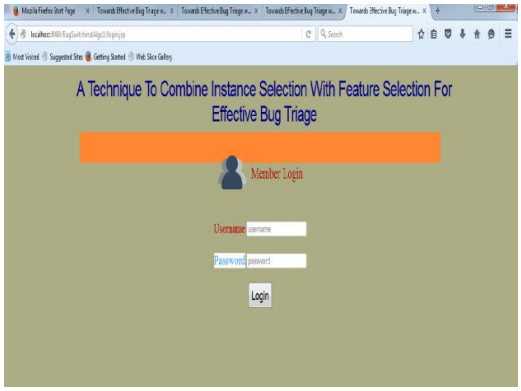

Fig.3 Member Login Page

The admin do the instance selection .In this module the no. of redundant bugs are removed. It works on rows. Figure 4 shows the instance selection module.

Fig. 4 Instance Selection

In this module also the duplicate records are removed but it works on the column. Both instance and feature selection reduce the bug data. Figure 5 shows the feature selection module.

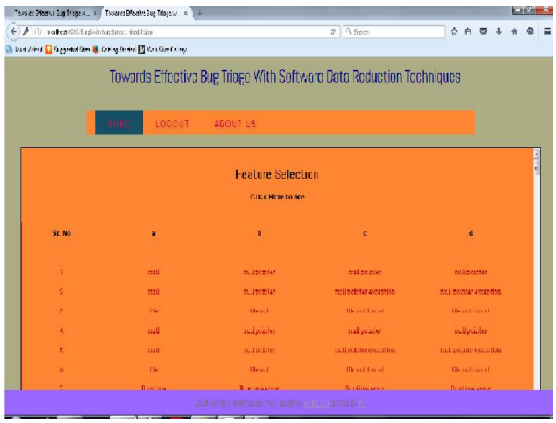

Fig. 5 Feature Selection

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

VIII. CONCLUSION

In the process of bug fixing bug triage is an important step. The aim of bug triage is to assign a new coming bug to the correct potential developer.It is an expensive step of software development and maintenance. In this paper, the size of bug data set is decreased by combining feature selection with instance selection and additionally enhances the quality of bug data. The order of applying instance selection and feature selection for another bug data set is decided by extracting the attributes of data set and prepare a predictive model based on previous data sets. My work provides an approach to form reduced and high-quality bug data in software development and maintenance.

REFERENCES

1. 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.
2. D. Cubrani_c and G. C. Murphy, "Automatic bug triage using text categorization," in Proc. 16th Int. Conf. Softw. Eng. Knowl. Eng., Jun. 2004, pp. 92–97.
3. 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.
4. 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.
5. 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, pp. 1–10.
6. 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.
7. T. Zimmermann, R. Premraj, N. Bettenburg, S. Just, A. Schr€oter, and C. Weiss, "What makes a good bug report?" IEEE Trans. Softw. Eng., vol. 36, no. 5, pp. 618–643, Oct. 2010.
8. 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.
9. C. Aggarwal and P. Zhao, "Towards graphical models for text processing," Knowl. Inform.Syst. vol. 36, no. 1, pp. 1–21, 2013.

BIOGRAPHY

Miss. Suvarna Kale receivedthe B.E. (Computer Engineering) from University of Pune in 2011 and pursuing M.E. degree in Computer Engineering from Institute of Knowledge College of Engineering, Savitribai Phule Pune University, Pimpale Jagtap Pune, and Maharashtra, India in 2015-16.

Mr. Ajay Kumar Gupta is working as Assistant Professor in Institute of Knowledge of COE, Savitribai Phule Pune University, Pimpale Jagtap, Pune and Maharashtra, India.