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 11, November 2016

Examine Effective Bug Triage with Software Data Reduction Techniques

Jam Jam Satyanarayana

M.Tech, Dept. of Computer Science and System Engineering with a Specialization in Bioinformatics,

Andhra University(Autonomous), Visakhapatnam, Andhra Pradesh, India

ABSTRACT: In this paper, we deal with the software bugs where large software company spent lot many of their cost in the same. The step of fixing the bug is called as bug triage where we correctly assign a developer to a new bug. Here, we address the problem of data reduction for bug triage. The problem of data reduction deal with how to reduce the scale and improve the quality. Hence, we combine instance selection with feature selection both simultaneously to reduce bug dimension and word dimension. We also extract the historical bug data set and predictive model to build new data set. This work provides leveraging techniques on data processing for high quality bug data in the software development.

KEYWORDS-bug triage, bug repositories, bug data reduction, feature selection, instanceselection

I. INTRODUCTION

In current world, software companies maintains a large-scale databases for storing the output of the software project, e.g., source code, bugs, emails, and specifications. Conventional software analysis is not fully suitable for the large-scale and complex data in bug repositories. Hence, Data mining has a promising means to handle bugs. Mining repositories can uncover interesting information in software repositories and solve the real world software problems. A bug repository has a collection of bug repots that plays an important role in managing bugs. Software bugs are always happening and clearing bugs is costly in software development. Huge software projects maintain bug repositories to collect the information and to help developers to handle the bugs. A bug repository maintains a bug report, which records the description the bug and the updates about the status of bug that has to be fixed. A bug repository contains several types of bugs such as fault prediction, bug localization, and reopened bug analysis.

II. LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then next steps are to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

III. METHODS AND MATERIAL

A. EXISTING WORK

In traditional software development, bugs are triaged by human triager, the developer who triages the new bugs manually. Triaging huge number of bugs manually takes more time and cost. To overcome this problem, an automatic bug triage system is introduced in existing system. It uses text classification technique, in which each reported bug is assigned to a developer. Developer is mapped to the label of the document containing bugs that are to be resolved. Bug triage is then converted into a problem of text classification and bugs are automatically solved with text classification techniques, e.g., Naive Bayes. From the results of text classification, a human triager assigns new bugs by incorporating his/her expertise. In text classification techniques accuracy, can be increased by investigating some

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 11, November 2016

further techniques e.g., a tossing graph approach and a collaborative filtering approach. However, the techniques of automatic bug triage are blocked in bug repositories which are low in quality. As data are a kind of simple text data, the well-processed bug data has to be generated to facilitate the application.

B. PROPOSED SCHEME

In proposed system, the problem of data reduction for bug triage is addressed, i.e., how the bug data is reduced to bring down the labor cost of developers and the quality is improved to facilitate the process of bug triage. Data is reduced by removing bug reports and words, which are redundant or non-informative. Bug triage aims to build an efficient set of bug data .In our work, the bug dimension and the word dimension are reduced simultaneously by combining the techniques instance selection and feature selection. Thus the reduced bug data has lesser number of

bug reports and lesser number of words than the original bug data. Although data is reduced it provides similar information as it is in the original bug data. The results of four instance selection algorithms and four feature selection algorithms are examined to avoid the bias of a single algorithm .when an instance selection algorithm and a feature selection algorithm is given, the order of applying these two algorithms may affect the results of bug triage.

In the proposed paper, predictive model is used to determine the order of applying instance selection and feature selection. From the experiments conducted over bug repots, it is identified that applying instance selection technique to the data set can reduce relevant subset of bug reports but the accuracy of bug triage may be decreased; applying the feature selection technique can reduce subset of relevant words in the bug data and the accuracy can be increased. Hence it is found that combining both these techniques can increase the accuracy, as well as reduce bug reports and words.

Contributions of this paper are as follows:
a) To simultaneously reduce the scales of the bug dimension and the word dimension
b) To improve the accuracy of bug triage.
c) Combination approach is proposed to address the problem of data reduction. That is application of instance selection and feature selection in bug repositories.
d) A binary classifier is built to predict the order of applying instance selection and feature selection.

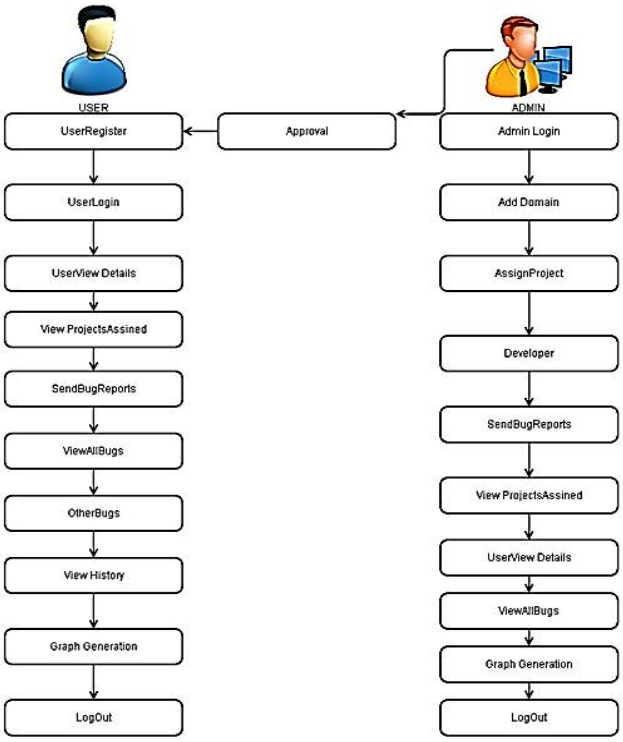

Figure 1. Proposed System

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 11, November 2016

C. BACKGROUND AND MOTIVATION BUG TRIAGE

A bug tiage is a formal process where each bug is priotitized based on its severity, frequency, risk and etc. The bugs are sorted based on the priority. Bugs with high priority will be fixed first. Low priority bugs will be fixed later. To achieve better balance working, important bugs are prioritized first. Three ways of differentiating bugs:

a) bug which is fixed.
b) bug which is fixed later and
c) bug which is not fixed.

A bug triage meeting for developers is held regularly for discussing about project life cycle. The Quality Assurance lead calls these meetings. The number of occurrences of meeting will vary from project to project, based on current status of project.

Motivation

Real-world data consists of noisy and redundant data which leads to increase in the cost factors of data processing. In the bug repository, all the bug reports are filled by developers in normal language. The bugs which are low in quality are collected in the bug repositories with the growth in scale. Such data which are inefficient may worsen the accuracy of fixing bugs.

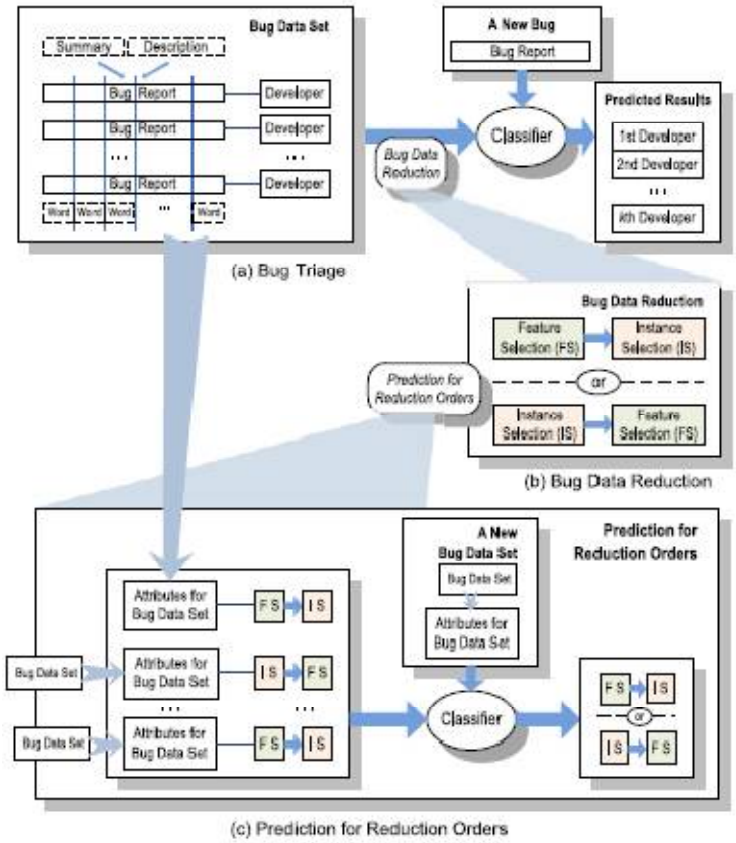

Figure 2. illustration of reducing bug data for bug triage

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 11, November 2016

IV. RESULT AND DISCUSSION

A. EXPERIMENTAL WORK

Data preparation

Data preparation means manipulating data into a form suitable for further analysis. Data preparation is the process of collecting, cleaning, and consolidating data. It is a process of pre-processing data.

Here, we present the data preparation for applying the bug data reduction. Bug triage predicts developers to fix bugs. The unfixed bugs are assigned to a developer to fix it. In bug repositories, some registered developers may have fixed very less number of bugs. Such in active developers who fixed less than 10 bugs are removed. Applying Instance Selection and Feature Selection.

In our work, we combine instance selection and feature selection to perform data reduction with more accuracy. The original data set is replaced with the reduced data set for bug triage. Instance selection and feature selection are widely used techniques in data processing. Instance selection is to obtain a subset of relevant instances (i.e., bug reports in bug data) while feature selection is to obtain a subset of relevant features (i.e., words in bug data). Instance selection technique reduces the number of instances by removing noisy and redundant instances. An instance selection algorithm can provide a reduced data set by removing relevant instances. Feature selection is a pre-processing technique for reducing relevant features for big scale data sets. When an instance selection algorithm IS and a feature selection algorithm FS is given, the order of applying has to be predicted. We use FS->IS to denote the bug data reduction, which first applies FS and then IS, and in IS->FS first applies IS and then FS.

Algorithm 1. Data reduction based on FS → IS

Input: training set $\mathcal{T}$ 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 $\mathcal{T}_{FI}$ for bug triage

1) apply FS to n words of $\mathcal{T}$ and calculate objective values for all the words;
2) select the top n_F words of $\mathcal{T}$ and generate a training set $\mathcal{T}_F$;
3) apply IS to m_I bug reports of $\mathcal{T}_F$;
4) terminate IS when the number of bug reports is equal to or less than m_I and generate the final training set $\mathcal{T}_{FI}$.

B. BENEFITS OF DATA REDUCTION

Reducing the Data Scale

Bug dimension: Bug triage is to assign developers for bug fixing. Developers can examine history of fixed bugs to find a solution to the current bug report. If the bug is already fixed then it can be replaced. The labor cost of developers can be saved by fixing bugs from history using instance selection, instead fixing in their own manually. The reduced data set can be handled more easily by automatic techniques (e.g., bug triage approaches) than the original data set.

Improving the Accuracy

Bug dimension: Instance selection can remove uninformative bug reports, while the accuracy may be decreased by removing bug reports. Word dimension: feature selection removes uninformative words, but the accuracy of bug triage is improved. This can recover the accuracy loss by instance selection.

V. CONCLUSION

A bug triage is a process done by software companies in order to maintain the bugs in their work process. In this project we use techniques such as instance selection and feature selection in order to improve the efficiency and maintenance of the bug. Thus the redundancy in the bug data set will be removed. It also helps us to assign the correct

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 11, November 2016

the correct project to the correct developer .The developer fixes less than 10 bugs will be removed by the Quality analyst (QA) during the bug triage process meet. **A SurveON**

TABLE I. COMPARISON OF VARIOUS BUG TRIAGING TECHNIQUES

S. N.	Title	Year	Techniques used for Bug Triage	Pros	Cons	Analysis
1	Automatic bug triage using text categorization	2004	Naïve Bayes Classifier	The bug can be automatically assign to the potential developer for evaluating all the bug report carefully which saves resources used in bug triage or bug assigning task.	The two problems with this approach is sometime the developer who fix the bug is not the one to whom it was officially assigned, second the algorithm does not proved to be as efficient as it was thought to be.	In order to improve the approach for handling bug triage is that on should involve the project's developer too. We should add on factors that would even deal with unlabeled document in corpus.
2	Who should fixed bug	2006	SVM, Naïve Bayes Classifier	A. It has provided help triager to assigning bugs more efficient. B. If company has little knowledge then new triager can work on it.	The processes only work on Eclipse, Firefox, and gcc; it does not work on other projects.	In order to analyzed that the improvement of the bug assignment process and it is found that gcc project is far worse than eclipse and firefox projects.
3	Modeling Bug Report Quality	2007	Bugzilla	A. It is provide to reducing overall software maintenance cost. B. Usage of this model leads to better Precision and recall result.	This process only work on two projects i.e., Mozilla and Firefox.	The reporter should be provided with Complete guidelines on what all information, attachment they should provide along with the bug report.
4	Information Needs in Bug Reports: Improving Cooperation	2010	Card sorting technique	In this paper, interaction between developer and user are necessary for fixing the bug in time.	The drawback is that, the result is only applicable for eclipse and Mozilla and might not be other projects.	It is analyzed that, it show the information status of bug report that means if any information is missing to filled by user then it
	Between Developers and Users					will show a "pending status" and will keep on notifying the reporter about it.
5	Towards Training Set Reduction for Bug Triage	2011	feature selection algorithm X 2 -test (CHI), instance selection algorithm, Iterative Case Filter (ICF)	Reduce the large scale of training set and decreases the noisy and redundant data in bug triage.	The disadvantage is that the combination is limited for each algorithms	We analyzed that, to reduce training set by using combination of instance selection and feature selection algorithm.
6	Efficient Bug Triaging Using Text Mining	2013	Naïve Bayes Classifier, five selection methods areLOR,X2, TFRF, MI, DFS	Automatic assign bug to the potential developer.	Cost increases due to the overloaded work distribution using Naïve Bayes algorithm.	It predicts an experienced developer to fix a new reported bug and redistribute the load of overloaded developer. And also analysed x2 selection mehod is more effective than other for bug assignment.
7	An Approach to Detecting Duplicate Bug Reports using N-gram Features	2014	N-gram Features, Cluster Chrinkage Technique	The technique has provided to improve the Detecting Duplicate Bug performance.	This technique is applicable for only AgroUML, Apache and SVN and might not applicable for Eclipse	We analyzed here,improvement of classification power for duplication detection by N-gram Features and divergence

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 11, November 2016

	and Cluster Chrinkage Technique				and Firefox	problem occurs due to N-gram Features is reduced by the Cluster Chrinkage
8	Bug Triaging: Profile Oriented Developer Recommendation	2015	Domain Mapping Matrix (DMM)	Rather than using historical bug report it uses domain mapping matrix for expertise profile of developer maintenance.	The drawback is only chroming bug repository is used.	It is analyzed that, To recommend the most suitable developer list for new bug reports.

REFERENCES

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. 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. 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. C. C. Aggarwal and P. Zhao, "Towards graphical models for text processing," Knowl. Inform. Syst., vol. 36, no. 1, pp. 1–21, 2013.
5. Bugzilla, (2014). [Online]. Avaialble: http://bugzilla.org/
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. 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. 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. 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. V. Bolon-Canedo, N. S anchez-Marono, and A. Alonso-Betanzos, "A review of feature selection methods on synthetic data," Knowl. Inform. Syst., vol. 34, no. 3, pp. 483–519, 2013.
11.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.
12.I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," J. Mach. Learn. Res., vol. 3, pp. 1157–1182, 2003.