ISSN 2395-1621

Bug treatment automation with the help of instance and feature selection using information security and data mining

[#1]Danish Shaikh, [#2]Alfaz Shaikh, [#3]Azharuddin Shaikh, [#4]Ivan Paul
[#5]Prof. Seema Vanjire

[4]paul_ivan52@yahoo.com

[#12345]Department of Computer Engineering,

Sinhgad Academy of Engineering, Pune.

ABSTRACT

When Software after it has been developed or in the intermediate phase the main issues are the bugs that occur and can be the reason of stoppage in functioning, degradation in performance, etc. of the developed product. In whole of the development phase of a product testing is one of the important phases which play a vital role in building and delivering a good quality product. It is said that Software cannot be 100% defect free by Exhaustive testing principle some of the bugs still exist. In Software Companies a huge amount of time and expenses are invested for solving reported bugs. Many of the company do the distribution of bugs to respective developer by manual means which can be error prone, time consuming and tedious job. In this project our aim is to manage the reported bugs through automation, which allocate the bug reports to the prior developer automatically. Some of the developer is experts in GUI related bugs whereas some are great expertise regarding to Java Applications. So to allocate the exact bug report to its expertise one through bug treatment system this idea is being evolved using two algorithms instance selection and feature selection that will help in classifying the bugs.

Keywords- Text mining, textual classification, software repositories, Data mining using information security, triaging, feature extraction.

ARTICLE INFO

Article History

Received: 9th November 2017

Received in revised form :

9th November 2017

Accepted: 11th November 2017

Published online :

12th November 2017

I. INTRODUCTION

Data mining is acquiring useful information from the given data. It's also known as knowledge discovery. Useful knowledge can be obtained as a result of data mining which can be used to reduce costs and increase income .Mining purpose uses two types of data, categorical and numerical .Data mining can only be done on data that is numerical or categorical. Many times, enterprise data found is non-numerical and non-categorical. For a successful business, extracting knowledge from this unstructured data can be time consuming therefore it has to be processed using different mining techniques like text mining techniques .Text mining uses techniques like Information extraction, information retrieval and natural language processing techniques. Classification is defined as a process of assigning items in a collection to predefined classes. The basic goal of classification is the accurate prediction of target class for each case in data. Status monitoring is used to determine when the bug is determined by the developer and when developer initiates its work on it and by what time the developer will complete with bug fixing .This will help to examine the status of bug and the condition through which it is going, but it will also help to earn some respect for the developer regarding his work.

II. MINING SOFTWARE REPOSITORIES

Software systems can be documented by artifacts called software repositories which often contain data from years of development of a software project. A software system is a very tiring task.

For example:

1. Runtime Repositories: Large organized storages which contain development logs about application usage on deployment sites and useful information of its execution are one of examples of runtime repositories.
2. Code Repositories: Examples of code repositories are Google code and codeforge.net which store source code of various open source projects.

3. Historical Repositories: Bug databases, source code repositories and archived communication logs are some examples of historical repositories.
4. Code Repositories: A process of software repository analysis that does discover significant and interesting information hidden in these repositories is known as MSR.

III. PROBLEM ESTABLISHMENT

Decide the order of treatment of bugs and assigning a developer to fix them is a seeming difficult to deal with the prospect and time consuming task. Developers, generally, need to be expert in some certain area. For example few developers could be expert in GUI while some could be in pure java functionality and many more task; hence assigning a specific bug to appropriate developer could save time. It can also help to maintain the interest level of the developers by assigning bugs according to their interest. It's not an easy task to assign right bug to the right developer for sorting of and allocation of bugs without knowing the actual class a bug belongs to. Method that differentiate open source software bugs using the summary and description of the bugs provided by the bug reporters is proposed in this research.

IV. LITERATURE SURVEY

[1]Reference 1, paper tells us about the two important algorithms, Instance and Feature algorithms that deals with mining repositories. These both algorithms are combination of multiple algorithms that we can use for our mining process. Naïve Bayes will be used for performing text classification.

[2]Reference 2, paper let us know that the source files happen to be in large contain compared to actual bug files present in it. Bug files could be way much smaller compared to whole source files. Hence, ranking approach has been implemented to source files to rank them according to their relevance.

[3]Reference 3, paper presents a framework for automatic assignment of bugs to Developers. They have explained, how using a software or an intelligent system you can assign bugs. They have also made use of Vector Space model to refer to the developers history of previously solved bugs. The vector model is used to summarize the data which can later be used to find similar reports by mining the data in the bug repository. So to create an efficient bug triage model , the author conducted meetings with the developers where in they enquired about their difficulties faced during solving bugs and many other information related in handling bugs. The author asked them about previous bug fixing experience, their satisfaction with the bug assignment, whether they were confident in handling bugs in the past. The information collected gave them an approximate estimates about the proposed model. This in turn helped them to implement the specimen model and test it within a software team working on the maintenance activities.

[4] Reference 4, Author found a technique by making use of data mining techniques to differentiate bugs of web based applications. It uses prediction of the bug type, it also uses debug strategy association rules which find the relationship between bug types and bug fixing solutions. The debug strategy method helps us with the part which consists of errors from the source code. After finding the errors, it becomes easier for the developer to fix them. The determined association rules help to predict files that usually change together such as functions and variables.

[5] Reference 5, Software Analysis and Intelligence Lab (SAIL) presented a brief history of the MSR field and discussed several recent achievements and results of using MSR techniques to support software research and practice. Author then explored the various opportunities and challenges that lie in the road ahead for this important and emerging field while highlighting work done in the MSR community to address some of these challenges.

[6]Reference 6, An algorithms to automatically and accurately identify bug-introducing changes. To remove false positives and false negatives by using annotation graphs, by ignoring non-semantic source code changes, and outlier fixes. Additionally, author validated that the fixes they used are true fixes by a manual inspection. Altogether, their algorithms can remove about 38%~51% of false positives and 14%~15% of false negatives compared to the previous algorithm. Finally, they show applications of bug-introducing changes that demonstrate their value for research.

[7] Reference 7, In this paper the author has brought up an evolutionary approach to automate the task of fixing bugs. The concept is to bring up the programs with a function that is based on number of unit tests they are able to pass. If a basic specification of the product having bugs is given then more so worldly fitness functions can be designed. Moreover, by utilizing the basic specification as an oracle, It can initiate number of unit tests as many we want. Hence, a co-evolution between programs and unit tests might take place to get better results. It is important to know that, to fix the bugs in a program with this novel approach, a user needs only to provide either a formal specification or a set of unit tests. No other information is required.

V. PROBLEM SOLVING

To get the solution for the bugs that are generated, we can implement our two important algorithms, instance selection and feature selection. These algorithms will help to reduce the amount of time taken for resolving the generated bugs.

VI. INPUT DATA

The data of software like Eclipse and Mozilla Firefox happen to be obtained from bugzilla -an open bug repository. Datasets of bug reports are obtained. This data is divided into training and testing groups, experiments are performed on different set of data from these groups. In this Project, we are using our Bug Repository and we will also make our own compiler for taking input. First we have to compile file and take input from there.

VII.MODEL FOR PREVISION

When a bug is generated, bug repository is used to store them and then the bug reports are obtained and it is submitted to our proposed system as shown in Fig. 1. Then the bug reports are pre-processed by system and extract all the terms in these reports with the help of bag of words approach. The terminology is of extremely high Depth and thus numbers of features have been reduced with the help of feature selection algorithm. These features have been used for training of classification algorithm which is used for classification of bug reports.

VIII. PRE-PRECISING

Data pre-processing is the most important step of data mining. Raw data can be obtained from bug repositories which are indirectly used for training the classification algorithm. Hence, the data needs to be pre-processed to make it working for training purpose. Data pre-processing is banausic step of data mining and it is important. Stop-words dictionary and regular expression rules are used to strain needless and not related words and filter the punctuations respectively. Stemming algorithm is used to stem the vocabulary.

IX. FEATURE SELECTION

After implementing bag of words approach on data, the vocabulary that is obtained has very large dimensionality.
Many of these dimensions are not related to text categorization and thus in turn result in reducing the attainment of the classifier. Feature selection is the process that helps to decrease the dimensionality of the obtained vocabulary. In this technique, best k terms out of the whole vocabulary are elected which contribute to accuracy and efficiency. There are a number of feature selection techniques such as Document Frequency (DF), Chi-Square Testing, Information Gain (IG), Term Frequency Inverse Document Frequency (TFIDF). In this research, we will use feature selection algorithm. Algorithm and its brief explanation: Applied to bugs obtained recently: IG, CH, SU, RF.

DESCRIPTION:
Input: Create new department and new bug pattern.
Output: Generate new department in database and define new bug pattern in new department.
D (I): Instant selection department
D (F): Feature Selection
Department P (f): bug Pattern

Steps are as followed:
1. If department D (I) not exist;
2. Go to feature selection
3. Create D (F) and send to database;
4. Define bug pattern p(f)
5. P(f) should be save in new D (F)
6. Terminate D(F);
7. Repeat 2 to 6 if again new department has to come.
8. End

X. INSTANCE SELECTION

Instance selection is another technique which is used for reducing the dimension of vocabulary obtained after applying bag of words approach. As most of the dimensions are related to our pre-defined bugs and result in reducing the performance of the classifier, hence to decrease the time, the process of Instance selection is used which chooses the best k terms out of the complete vocabulary which contribute to accuracy and efficiency. This selection is fast instance of feature selection. Algorithm and it's explanation: Applied sequentially: ICF, LVQ, DROP, POP

DESCRIPTION:
Input: training set T with n stop words and m bug report,
Reduction order IS -> FS
Final number n (f) of words, Final number m (I) of bug report,
Output: reduced, triage and send data set T (FI) to matching department.

Steps are as followed:
1. Admin (project manager) apply IS to n stop words T and calculate objective values for all words;
2. Select the top n (I) words of T and generate a training set T (I);
3. Filter bugs F (I) and send to suitable department D (I);
4. Terminate IS when new department to come;
5. End

XI. CLASSIFIER MODELING

An automated process to find some metadata about a document is considered as "Text classification." It has been used in various areas like document indexing by suggesting its categories in a "content management system", "spam filtering", "automatic help desk requests" sorting etc. For bug classification, Naïve Bays text classifier is used in this research. Naïve Bays classifier is based on Bays‟ theorem with maverick assumption and is a probabilistic classifier. It implies that the classifier speculates that any feature of a class is unassociated to the presence or absence of any other feature.

XII.BUG CLASSIFICATION SYSTEM

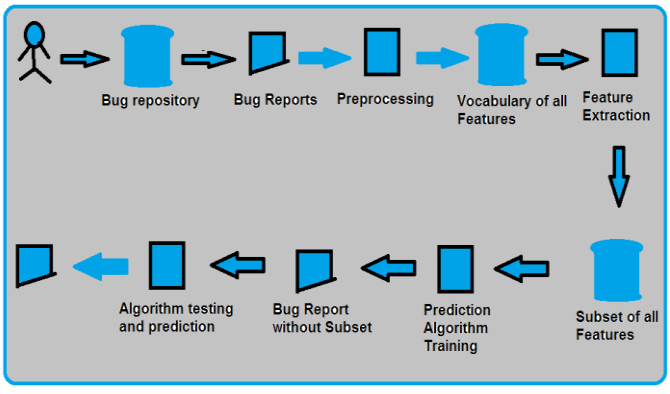


Fig. 1 Block diagram illustrating classification of bugs

XIII. ARCHITECTURE DIAGRAM

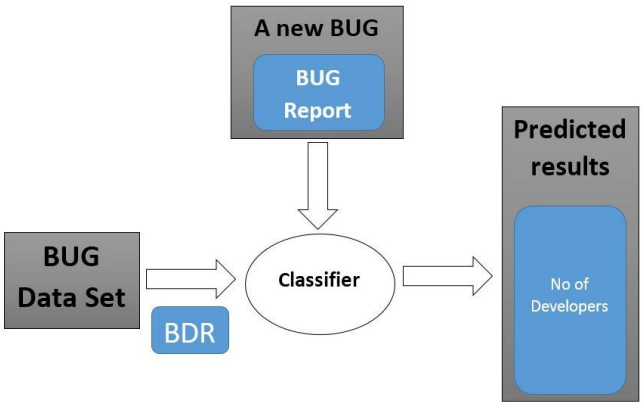
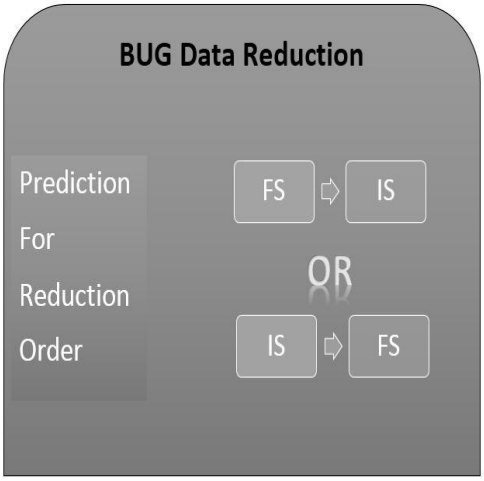


Fig. 2 System Block Diagram



FS = Feature Selection; IS = Instance Selection

Fig. 3 BDR(Bug Data Reduction)

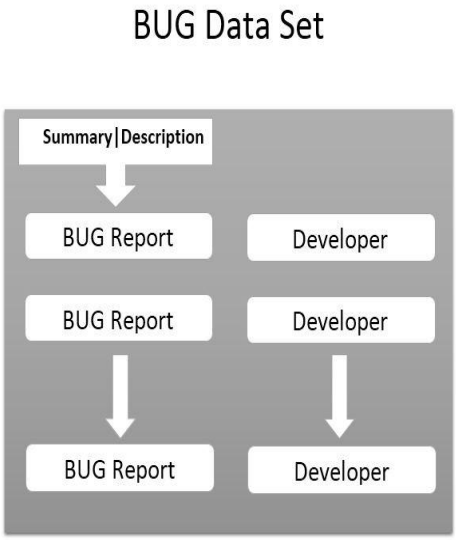


Fig. 4 Bug Data Set

Modules and Description

This project shall require the implementation of following

Modules: Authentication users, Products, Bug details, View, Admin and Logs.

Authentication users: The Bug Tracking System first activates the login form. Here the user is prompted to enter the „user name" and „password" and our system starts the authentication process in which the username and password are verified with the help of existing username and password in the database. If the password matches then it is allowed to the main page else it warns the user for Invalid User name and password. After successful authentication, the system activates menus. The activity log has also been prepared for failures.

Products: In order to complete the project each product is allocated with Resources. First of all employees' credentials like their names and qualifications are stored in the database. Each user is allotted to the product based on their rating, Qualification and designation. For each user, effective date which specifies the total period a user is valid for is stored. Also list of products, product versions and product details are stored for future convenience.

Bug Details:

The user is provided with the facility for adding bugs, updating the existing bugs in this module. As the number of bugs for a product can be very large, this system is equipped with efficient filtering. The user can filter the bugs based on priorities, databases, operating systems and status. After the user applies filter the list of bugs are displayed from the database. More modules might be added as and when needed in to the stage of project development.

XIV. CONCLUSION

In open source bug repositories, bugs have been reported by users. Triaging of these bugs is a repetitive and lasting for long time task. If some proper class is assigned to these bugs, then they could be easily allotted to a closely connected developer and thus bugs can be fixed efficiently. Still, as reporters of these bugs are mostly non-technical it would be impracticable for them to assign correct class to these bugs. In this research polynominal Naïve Bays text classifier is used for classifying software bugs. Instance selection algorithm and Feature selection algorithm are used for bug sorting of and allocation of treatment. Maximum exactness at divination can be obtained using this system. Bug triage is an costly step of software maintenance in both labor and time cost. In this project, advantages like reduction in bug data sets and improved data quality can be obtained because of mixture of instance and feature selection.

FUTURE WORK

The main provocation would be performing classification for numerous numbers of domains that could be uninteresting but important process. This will help teams of developers under one roof to work on their separate domains with fairness and utilization of time will be done satisfactorily.

REFERENCES

[1] Jifeng Xuan, He Jiang, Member, IEEE, Yan Hu, Zhilei Ren, Weiqin Zou, Zhongxuan Luo, and Xindong Wu, Fellow, IEEE" Towards Effective Bug Triage with Software

Data Reduction Techniques ” January 2015

[2] XinYe, RazvanBunescu, Chang Liu, Mapping bug reports to relevant files : A ranking model , a fine-grained benchmark and feature evaluation, IEEE , VOL. 42, NO. 4, APRIL 2016.

[3] Micheal W. Godfrey, Olga Baysal and Robin Cohen, A bug you like: A Framework for Automated assignment of bugs, IEEE, June 2016

[4] Lei Xu , Lian Yu , Jingtao Zhao , Changzu Kong , Huihui Zhang, “data mining techniques to differentiate bugs of web based applications”

[5] Ahmed E. HassanSoftware Analysis and Intelligence Lab (SAIL)School of Computing, Queen’s University, Canada, “The Road Ahead for Mining Software Repositories”,IEEE

[6] Sunghun Kim, Thomas Zimmermann, Kai Pan, E. James Whitehead, Jr,”Automatic Identification of Bug-Introducing Changes”,IEEE

[7] Andrea ArcuriThe Centre of Excellence for Research in Computational Intelligence and Applications (CERCIA),The School of Computer Science, The University of Birmingham, Edgbaston, Birmingham B15 2TT, UK.” On the Automation of Fixing Software Bugs”,ICSE

