ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 3, Issue 5 , May 2016

Bug Classification System Using Data Reduction Techniques

Monika Dangore, Joyce Silvera, Ekta Dhurmekar, Meghna Suri

Project guide, Department of Computer Engineering, DR.DY.PATIL SCHOOL OF ENGINEERING, Pune,India.
Student, Department of Computer Engineering, DR.DY.PATIL SCHOOL OF ENGINEERING, Pune, India.
Student, Department of Computer Engineering, DR.DY.PATIL SCHOOL OF ENGINEERING, Pune, India.
Student, Department of Computer Engineering, DR.DY.PATIL SCHOOL OF ENGINEERING, Pune, India.

ABSTRACT: Open source projects consist of open source bug repositories. User reports bugs to these repositories and they are usually non-technical and cannot assign correct class to these bugs. Triaging of bugs is a tedious and time consuming task. Developers are usually expert in specific areas. For example, some developers are connoisseurs of GUI and others in Java applications. Assigning a particular bug to respective developer could save time and would help the developers by assigning bugs according to their requirements. However, assigning correct bug to respective developer is quite difficult for triager without knowing the actual class, to which the bug belongs. In this paper, we have implemented the enlisted reference papers for bug triaging in which the various triaged bug reports are assigned to the respective developers using data reduction techniques.

KEYWORDS: Text mining, classification, software repositories, open source software projects, triaging, feature extraction

I. INTRODUCTION

The process of extracting useful information through data analysis is called data mining. It is also known as knowledge discovery. Useful knowledge is obtained as a result of data mining which can be used to cut costs, increase revenues or both. There are two types of data categorical and numerical used for mining purpose like integer, decimal, float, char, varchar2 etc.

Data mining cannot be done on data that is not numerical or categorical. Most of enterprise data found is non numerical and non categorical [1]. For the success of business, extracting knowledge from this unstructured data can be difficult therefore it is processed using text mining techniques so that it can be processed by data mining algorithms and techniques. Some techniques used for text mining are Information extraction, information retrieval and NLP (natural language processing) techniques.

The process of assigning items in a collection to predefined classes or categories is called Classification. Basic goal of classification is the accurate prediction of target class for each case in data. For example, loan applications can be classified into high, medium or low risks on the basis of classification model.

II. MINING SOFTWARE REPOSITORIES

To understand constantly evolving software systems is a very daunting task. History of software systems are maintained in software repositories. Evolution of software systems are documented by artefacts called Software repositories. They often contain data from years of development of a software project [2].

Examples of software repositories are:

Runtime Repositories: Repositories that contain development logs about application usage on deployment sites and useful information of its execution are one of many examples of runtime repositories.

Historical Repositories: Bug repositories, source code repositories and archived communication logs are some examples of historical repositories. Code Repositories: Examples of code repositories are Google code and codeforge.net that store source code of various open source projects [3].

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 3, Issue 5 , May 2016

A process of software repository analysis which discovers significant and interesting information hidden in these repositories is known as MSR. It processes and analyses the huge software engineering data to detect interesting patterns in this data. It is an open field as in what can be mined and what can be learned from practice. All kinds of software repositories can be mined.

III. PROBLEM FORMULATION

Triaging of bugs and assigning a developer to fix them is a tedious and time consuming task. Developers are generally expert in some certain area. For example few developers are expert in GUI while some are in pure java functionality etc; therefore assigning a specific bug to admissible developer could save time. It would also help to maintain the interest level of the developers by assigning bugs according to their interest. It is not an easy task to assign right bug to the right developer for tri-ager without knowing the actual class a bug belongs to. A technique for classification of 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

Following is an excerpt of the already implemented techniques for software bugs classification are:

A. Micheal W. Godfrey, Olga Baysal and Robin Cohen presented a framework for automatic assignment of bugs to developers for fixation using vector space model [4].

In this paper [4], the authors have proposed a specimen of the intelligent system that instinctively conducts the bug assignment. They have employed the vector space model to infer information about the developer's expertise from the history of the previously fixed bugs. The vector model is used to retrieve the title and the description from the report to build a vector which later can be used to find similar reports by mining the data in the bug repository. In order to create an efficient bug triage model, the authors conducted a survey wherein they collected a feedback from the developers regarding their previous bug fixing experience, their satisfaction with the bug assignment, whether they were successful and confident in handling bugs in the past, etc. The overall information provided them the initial estimates for 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.

B. Hemant Josh, Chuanlei Zhouang, Oskum Bayrak presented a methodology to predict future bugs using history data [5].

In this paper [5], the authors have presented a bug prediction algorithm, the purpose of which is to predict the number of bugs which are to be detected and reported in each month. So the bug prediction of any month basically depends upon the bug count of the precursor month. All this prediction is achieved through the prediction-alogorithm implemented in the respective paper[5].

C. Lei Xu, Lian Yu, Jingtao Zhao, Changzhu Kong, and HuiHui Zhang proposed a technique by making use of data mining techniques to automatically classify the bugs of web-based applications by predicting their bug type.

In this paper [6], the authors have put forth the debug strategy association rules which find the relationship between bug types and bug fixing solutions. The debug strategy acquaints us with the erroneous part of the source code. Once the errors are found then it is very easy for the developers to fix them. The determined association rules help to predict files that usually change together such as functions or variables.

D. Nicholas Jalbert and Westley Weimer proposed a system that checks the duplicity in the bug reports.

In this paper[7], the authors have instantiated a model that automatically indicates whether an arriving bug report is original or duplicate of an already existing report. It saves the developer's time and efforts. To predict bug duplication, system they have made use of rudimentary methods such as textual semantics, clustering in graphs and surface features.

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 3, Issue 5 , May 2016

V. PROBLEM SOLUTION

This section describes the proposed system for bug classification, data used for classification task and results obtained in different experiments.

VI. INPUT DATA

The data of software such as Eclipse and Mozilla Firefox is obtained from bugzilla -an open bug repository [9] [10]. Dataset of bug reports is obtained. This data is divided into training and testing groups and experiments are performed on different set of data from these groups. In Our Project We Are Using Our Bug Repository and we also make our own compiler for taking input. First we compile file and take input from there.

VII. MODEL FOR PREDICTION

When a bug is reported it is stored in the bug repository then the bug reports are extracted and it is submitted to our proposed system as shown in Fig. 1. System pre-processes the bug reports and extracts all the terms in these reports using bag of words approach. The vocabulary is of extremely high dimensionality and thus numbers of features are reduced by using feature selection algorithm. These features are used for training of classification algorithm which is then used for classification of bug reports. The classification algorithm used in proposed system is multinomial Naïve Bayes.

VIII. PRE-PROCESSING

The most important step of data mining is data pre-processing. Raw data is obtained from bug repositories which cannot be directly used for training the classification algorithm. Therefore the data needs to be pre-processed to make it functional for training purpose. Data pre-processing is a monotonous step of data mining and most important as well. Stop-words dictionary and regular expression rules are used to filter useless words and filter the punctuations respectively. Stemming algorithm is used to stem the vocabulary.

IX. FEATURE SELECTION

After applying “bag of words” approach on data the vocabulary obtained has very large dimensionality many of these dimensions are not related to text categorization and thus result in reducing the performance of the classifier. Feature selection is the process used to decrease the dimensionality of the obtained vocabulary. In this technique tthe best k terms out of the whole vocabulary are chosen which contribute to accuracy and efficiency.

There are a number of feature selection techniques such as Chi-Square Testing, Information Gain (IG), Term Frequency Inverse Document Frequency (TFIDF), and Document Frequency (DF). In this research, we are using feature selection algorithm.

X. INSTANCE SELECTION

Instance selection is another technique 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. Therefore to decrease the time, the process of Instance selection is used which chooses the best k terms out of the whole vocabulary which contribute to accuracy and efficiency. This selection is fast instance of feature selection.

XI. CLASSIFIER MODELING

An automated process of finding some metadata about a document is referred as Text classification. It is used in various areas like document indexing by suggesting its categories in a content management system, spam filtering, automatic help desk requests sorting etc. Naïve Bayes text classifier is used in this research for bug classification. Naïve Bayes classifier is based on Bayes’ theorem with maverick assumption and is a probabilistic classifier. It means the classifier assumes that any feature of a class is unassociated to the presence or absence of any other feature. [11]

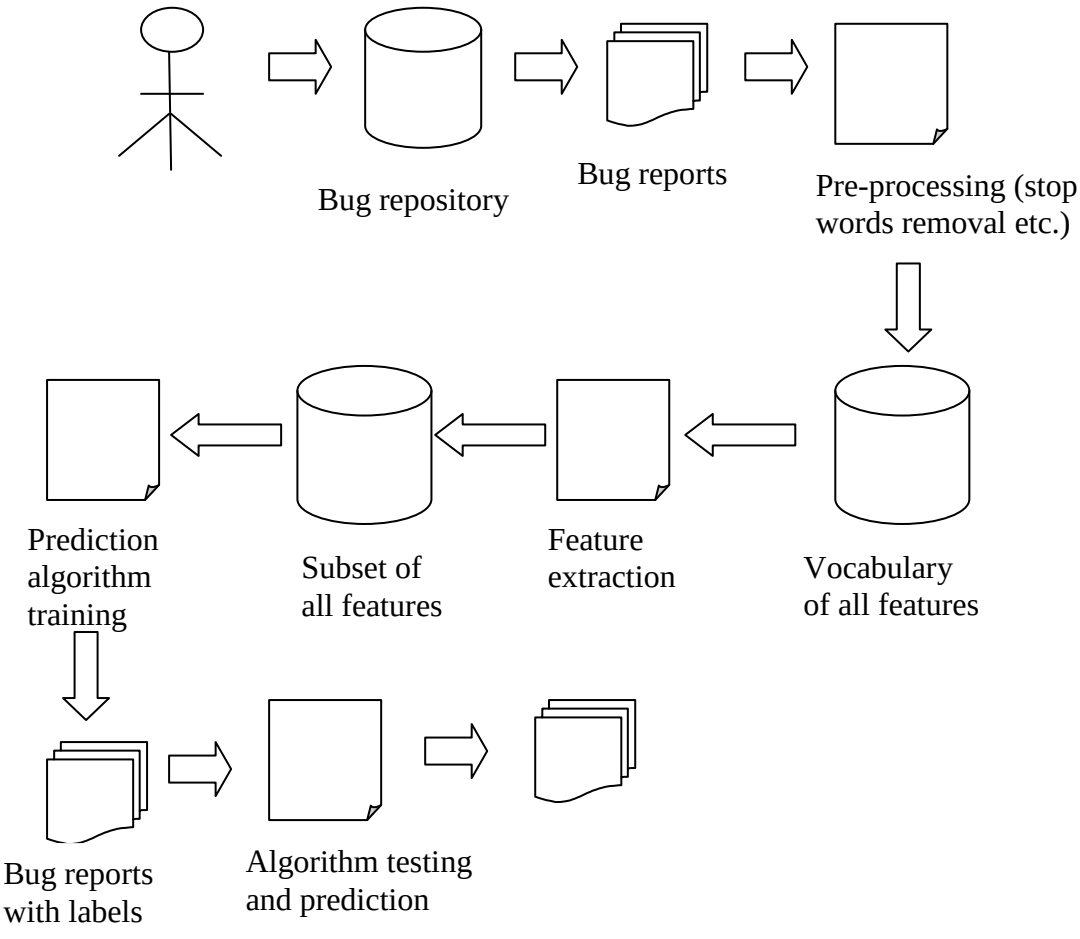
ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 5 , May 2016

XII. PROPOSED SYSTEM

Using this system, we can obtain maximum prediction accuracy in terms of both labor cost and time cost bug triage is an expensive step of software maintenance. In this project, we combine following advantages:
1. To reduce scale of bug data sets as well as improve data quality, we use feature selection with instance selection.
2. Another advantage is to determine order of applying instance selection and feature selection for a new bug data set.
3. We use our own bug's repository for training data set instant of bugzilla or other.
4. We use our own compiler for compiling and run file.

Compiler:
We created our own module called "Online java compiler with security editor" to ease our work. The main goal behind its creation is that we can easily write a java program and compile it and debug it online .The client machine is connected to the server only. The Client machine doesn't have the development kit called Java development kit (JDK). The server with java compiler executes the java code and produce error message to the appropriate client machine.

Java File Compilation:
With this module we can compile any created java application. There is no requirement to have JDK installed on client machines. The client using this web application can compile the java file. The client's java application is compiled by the JDK installed in the server machine. The JDK installed in the server machine compile all the java programs available in the server machines.



ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 3, Issue 5 , May 2016

XIII. EXPERIMENTS

A. Data Preparation

In this part, we present the data preparation for applying the bug data reduction. We evaluate the bug data reduction either on bug repositories or bugs provided by compiler. Our compiler can compile java program, dot net programs, C program, C++ programs etc.

- We use our own bug's repository for training data set instant of bugzilla or other (Eclipse and Mozilla). Eclipse is a multi-language software development environment, including an Integrated Environment (IDE) and an extensible plug-in system; Mozilla [33] is an Internet application suite, including some classic products, such as the Firefox browser.
- We use our own compiler for compiling and run file. First we compile file and take input from there.

B. Data sets and evaluation

To examine the results of data reduction, we will employ instance selection and feature selection algorithms. When a bug is reported it is stored in the bug repository then the bug reports are extracted and it is submitted to our proposed system as shown in Fig. 1. System pre-processes the bug reports and extracts all the terms in these reports using bag of words approach. The vocabulary is of extremely high dimensionality and thus numbers of features are reduced by using feature selection algorithm. These features are used for training of classification algorithm which is then used for classification of bug reports. The classification algorithm used in proposed system is multinomial Naïve Bayes. If we discover new error, it automatically gets stored in created bug database, so next time if we confront same error we can associate it with this bug database to increase efficiency.

XIV.RESULTS

Results are obtained on the basis of prediction accuracy. Prediction Accuracy is defined as
"Ratio of the bug reports with correct class to the total
Number of bug reports."

Experimental results showed that instant selection gives better results in case of bug classification from open source bug repositories. Instant Selection gives higher accuracy as compared to feature selection with same testing to training ratio. Proposed system using Naïve Bayes classifier is a probability based approach that works on the prior Probability of classes and conditional probability of features in the classes. Another important model for text classification is support vector machine Lian Yu, Lei Xu, HuiHui Zhang and Jingtao Zhao[6] used SVM for bug classification. Proposed technique using naïve Bayes text classifier has following advantages over the proposed system:

- When training data is small, proposed system performs better than SVM based system of Lei Xu [6]. Training curve for SVM is much greater than Naïve Bayes and when enough training set is not given it does not perform well.
- As far as processing time is concerned Lei system is in a disadvantage. Processing time is much higher than the other proposed technique and it grows quadratial ly as the number of documents increases in training set.

XV. ALGORITHM

Instant selection algorithm 1:
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

Admin (project manager) apply IS to n stop words T and calculate objective values for all words;
Select the top n(I) words of T and generate a training set T(I);
Filter bugs F(I) and send to suitable department D (I);

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 5 , May 2016

Terminate IS when new department to come;
end

Feature Selection Algorithm 2:
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

If department D (I) not exist;
Go to feature selection
Create D (F) and send to database;
Define bug pattern p(f)
P(f) should be save in new D (F)
Terminate D (F);
Repeat 2 to 6 if again new department has to come.
End

XVI. FUTURE WORK

The main challenge to address in the future is the strategic developers. In time, the developers could learn how the system assigns bug fixing tasks and try to manipulate task assignment. Thus, we should clinch that assignment of bugs is a fair and manipulation-free process.

XVII. CONCLUSION

It is known that bugs are reported by users in open source bug repositories. Triaging of these bugs is a monotonous and protracted task. After assigning some proper class to these bugs, they could be easily assigned to a relevant developer and in a way bugs can be fixed effectively and efficiently. Since the reporters of these bugs are mostly non-technical so it would be difficult for them to assign correct class to these bugs. In this research an automated system for classifying software bugs are formulated and classifier used is multinomial Naïve Bayes text classifier. We have used two algorithms for bug traige, one is Feature selection algorithm and another is instant selection.

REFERENCES

[1] A. Hotho, A. Nürnberger and G. Paaß, "A Brief Survey of Text Mining," vol. 20, GLDV Journal for Computational Linguistics and Language Technology, 2005, pp. 19-62.
[2] A. E. Hassan, "The Road Ahead for Mining Software Repositories," *IEEE Computer society*, pp. 48-57, 2008.
[3] S. Diehl, H. C. Gall and A. E. Hassan, "Special issue on mining software repositories," in *Empirical Software Engineering An International Journal © Springer Science+Business Media*, 2009.
[4] O. B. Michael and G. C. Robin, "A Bug You Like: A Framework for Automated Assignment of Bugs.," *IEEE 17th international conference*, 2009.
[5] C. Zhang, H. Joshi, S. Ramaswamy and C. Bayrak, "A Dynamic Approach to Software Bug Estimation," in *SpringerLink*, 2008.
[6] L. Yu, C. Kong, L. Xu, J. Zhao and H. Zhang, "Mining Bug Classifier and Debug Strategy Association Rules for Web-Based Applications," in *08 Proceedings of the 4th international conference on Advanced Data Mining and Applications* , 2008.
[7] N. Jalbert and W. Weimer, "Automated Duplicate Detection for Bug Tracking Systems," in *IEEE computer society*, 2008
[8] T. Bruckhaus, C. X. Ling, N. H. Madhavji and S. Sheng, "Software Escalation Prediction with Data Mining," in *Data Mining, Fifth IEEE International Conference*, 2006.
[9] [Online]. Available: https://bugzilla.mozilla.org/.
[10] [Online].Available:https://bugs.eclipse.org/bugs/.
[11] [Online].Available:https://en.wikipedia.org/wiki/Naive_Bayes_classifier .