ISSN 2395-1621

Classification Of Bugs With The Help Of Instance And Feature Selection On Windows Platform

[#1]Shoeib Khan, [#2]Pankaj Karad, [#3]Harsh Vardhan, [#4]Vivek Singh,
[#5]Mrs.Urmila Biradar

[1]shoaibkhan136@gmail.com
[2]pankajkarad0099@gmail.com
[3]hvrocker@gmail.com
[4]vivekkun@gmail.com
[5]urmila.biradar@raisoni.net

G.H Raisoni College of Engineering And Management Wagholi
Pune, India

ABSTRACT

Open source projects such as Eclipse and Firefox 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 triaged without knowing the actual class, to which the bug belongs. In this paper, we have surveyed the enlisted reference papers for bug triaging in which the various triaged bug reports are assigned to the respective developers using data reduction techniques.

KEY WORDS: Text mining, classification, software repositories, open source software projects, triaging, feature extraction.

ARTICLE INFO

Article History

Received: 8th June 2017

Received in revised form :

8th June 2017

Accepted: 10th June 2017

Published online :

10th June 2017

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. 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.

Status monitoring can be applied to determine when does the bug arrive to developer and when does developer actually start working on it and by what time the developer will be done with fixing this particular bug.

This will help to monitor not just status of bug and the condition through which it is going , it will also help to earn some fairness for the developer as his credentials will be generated based on this approach.

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 artifacts called Software repositories. They often contain data from years of development of a software project .

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 .

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:

1) Towards effective bug triage with software data reduction techniques
Jifeng Xuan , He Jiang , Yan Hu , Zhilei Ren , Weiqin Zou , Xindong Wu
This 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) Mapping bug reports to relevant files : A ranking model , a fine-grained benchmark and feature evaluation
Xin Ye , Razvan Bunescu , Chang Liu
This 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) Micheal W. Godfrey, Olga Baysal and Robin Cohen presented a framework for automatic assignment of bugs to developers for fixation using vector space model .

In this paper, 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

4) Lei Xu , Lian Yu , Jingtao Zhao , Changzu Kong , Huihui Zhang proposed a technique by making use of data mining techniques to automatically classify bugs of web-based applications by predicting their bug type
The authors have put forth the bug the debug strategy association rules which find the relationship between bug types and bug fixing solutions. The debug strategy acquaints us with the errorneous 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 ususally change together such as functions and variables.

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's such as Eclipse and Mozilla Firefox is obtained from bugzilla -an open bug repository . 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 the 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, automatica 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 maveric 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]

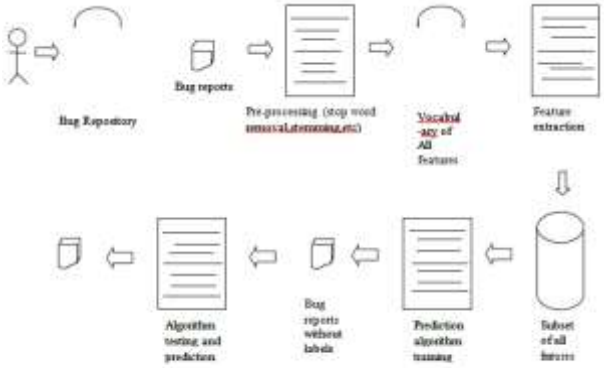

Fig.1: BUG CLASSIFICATION SYSTEM

XII. CONCLUSION

In open source bug repositories, bugs are reported by users. Triaging of these bugs is a monotonous and protracted task. If some proper class is assigned to these bugs then they could be easily assigned to a relevant developer and thus bugs can be fixed effectively and efficiently. However, as reporters of these bugs are mostly non-technical it would be unfeasible for them to assign correct class to these bugs. In this research an automated system for classifying software bugs is formulated, using multinomial Naïve Bayes text classifier. Feature selection algorithm and instant selection algorithm are used for bug triage. Maximum prediction accuracy can be obtained using this system.
Bug triage is an expensive step of software maintenance in both labor cost and time cost. In this project, we combine following advantages:

1. We use feature selection with instance selection to reduce the scale of bug data sets as well as improve the data quality. To determine the order of applying instance selection and feature selection for a new bug data set.

2. We use our own bug's repository for training data set instant of bugzilla or other.

3. We use our own compiler for compiling and run file.

XIII. FUTURE WORK

The main challenge would be performing classification for numerous numbers of domains that could be tedious

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 properly.

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_classifie

