TWO WAY CLASSIFICATION APPROACH OF STATISTICAL BUG REPORTS

Pankaj Kumar\textsuperscript{1}, Mr. Nishant Anand\textsuperscript{2}

\textsuperscript{1}Student, CSE Department, CBS Group Of Institutions, Jhajjar, Haryana

\textsuperscript{2}Professor and Head of CSE Department, CBS Group Of Institutions, Jhajjar, Haryana

Abstract—It is very important to analyze the bugs that is collect by various sources (like from users, developers that submit the bug report and testers that classified the bug reports) into right categories. In this paper we introduce a bug tracking system tools that analyze the bugs in two different ways. That help to classify bug reports. In first ways we introduce Naïve Bayes classification process by which we find probability of bugs categories on the basis of attributes of category of bug dataset. And in second ways we use natural language processing in summary attributes of bug dataset. By compare the results of two methods we able to classify more accurately and efficiently.

Keywords—Bug Tracking System, Natural language processing, Stemming, Naïve Bayes classifications, SBR, NSBR.

INTRODUCTION

A software bug is an error, flaw, mistake, failure, or fault in a computer program or system that produces an incorrect or unexpected result, or causes it to behave in unplanned ways. Most bugs arise from mistakes and errors made by people in either a program’s source code or its design [1]. Bug Tracking System has life cycle as shown in following fig.

![Bug Tracking Life Cycle](image)

Figure 1: Bug Tracking Life Cycle [2]

Bugs move through a series of states in their lifetime. When a bug report is submit to the bug tracking system (BTS) its status is set to New Bug. Once a developer has been either assigned to or received responsibility for the report, the status is set to Assigned. When a report is closed its status is set to Resolved. It may further be marked as being verified or closed for fine. A report can be resolved in a
number of ways; the resolution status in the bug report is used to record how the report was resolved. If the resolution resulted in a change to the code base, the bug is resolved as fixed. When a developer determines that the report is a copy of an existing report then it is marked as duplicate. If the developer was unable to reproduce the bug it is indicated by setting the resolution status to works-for-me. If the report describes a problem that will not be fixed, or is not an actual bug, the report is marked as wont-fix or invalid, respectively. A formerly resolved report may be reopened at a later date, and will have its status set to re-opened [3].

**PROPOSED METHODOLOGY**

In this methodology we load the data from Bug tracking system database that have different attributes such as product, component, status, resolution, Summary and its category state. After this we preprocess the data set by removing null values and redundant data. After that we apply bayes’ rules for prediction. We considers only maximum probability of SBR(Security Bug Report) or NSBR(Non Security Bug Report). We out list minimum probability values. After gaining the probability we use natural language processing (NLP) on the summary attributes of data. We fetch the summary for surety of SBR we analyze bug report with natural language process on the summary of the reports. Now with the help of natural language processing we read line by line text and only consider security related terms like vulnerability, attack etc. We have to evaluate the bug related text and remove the stop words such as articles, prepositions, and conjunctions that are not use in text mining. For this we maintain stop words list. Get the maximum count for words occurrence. If the frequency of the words in whole dataset will give the maximum security related terms then it is SBRs and treats as SBRs. After this we Mach this result with bayes prediction if Mach then we confidentially say this reports is SBR. Following figure describes the proposed methodology.

**RESULTS**

Bug tracking dataset have different attributes such as product, component, status, resolution, Summary and its category stage.
We enter the value for product is Bugzilla, for component is Chatzilla, for Resolution is Incomplete then status is Resolved.

From above we get maximum probability for SBR is 2.79E-07, with respect to Bugzilla, chatzilla, Incomplete, and Resolved with respective to whole dataset.

After naïve bayes classifications we use natural language processing in which we read line by line text and only consider security related terms like vulnerability, attack etc. After processing we get the terms with the frequency, in grid forms. If the terms are security related terms then the bug report for corresponding attributes are SBR.
After natural language processing we get security related word in grid forms. We get the terms with the frequency, in grid forms. If the frequency of terms is security related terms then the bug report for corresponding attributes are SBR.

Following window analyze that our bug report is SBR or NSBR. But we get SBR its means that our bug category is SBR.
CONCLUSION

Current bug tracking systems do not effectively produce all of the information needed by developers. Without this information developers cannot resolve bugs in a timely method. We analyze the same bug report in summary attributes in different ways. First we analyze the bug with help of naïve bayes classification which is efficient to other classification algorithm. Natural language processing enables us to implements a more automated and more efficient bug triaging process. By use of this automated system the security engineers independently reviews each BR. If two security engineers disagreed on their evaluations of manually labeled BRs, then they discussed their differences and reached an agreeable consensus (result). By analyzing same report with different methods it increase the efficiency of the system. While implementing a range of improvements from these areas may be ideal, bug tracking systems may instead prefer to specialize, thus providing a rich set of choices. Finally we consider two mode of bug classification such as semantic based and attribute too, typically essential requirement from summary/comment point of view get fetch bug logs as needed to resolve for estimated product type in efficient time.

References

[3] John Anvik, Lyndon Hiew and Gail C. Murphy Department of Computer Science University of British Columbia “Who Should Fix This Bug?”

