ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 2, March 2016

MINIMIZING SOFTWARE BUGS USING DATA REDUCTION TECHNIQUES

K.NEETHU[#1], K.SRIDEVI[#1], V.VYSHNA[#1], Ms.I.SIBIYA[#2]
#1-Bachelor of Computer Science and Engineering
#2 Assistant Professor of Computer Science and Engineering
BHARATHIYAR INSTITUTE OF ENGINEERING FOR WOMEN

Abstract— In an bug tracking system, different testers need to submit more reports on the different bugs. Referred to as copy data's, this may cost extra maintenance efforts in recover. To minimize the time cost in physical work Text classification techniques are useful to conduct automatic bug triage. To reduce the scale and improve the quality of bug data using instance selection with feature selection. By applying this procedures, to extract attributes from historical error data sets and make a predictive model for a new bug data set. To identify duplicates accurately using recovery function. To evaluated three large software bug repositories from Mozilla, Eclipse and OpenOffice.The results of data reduction can effectively reduce the data scale and improve the accuracy of bug triage.

INTRODUCTION

Due to complexities of software and keep track of bugs and their associated fixes, bug tracking system like Bugzilla1 has been proposed and is widely adopted. Traditional software development, software repositories are large-scale databases e.g., source code, bugs, emails, and specifications. This software analysis is not completely suitable for the large-scale in software repositories. Data mining handle software data bug repository plays an important role in handling software bugs. Software companies spend over their time cost in fixing bugs. In a bug repository, a bug is maintained as a bug report, which records the textual explanation of reproducing the bug and updates near to the position of bug fixing. The bug reports in a bug repository are called bug data.

There are two problems related to bug data that are, the large scale and the low quality. Software techniques affect from the low quality of bug data. Two main characteristics of low-quality bugs are noise and redundancy.

Handling software bug is bug triage, to assign a correct developer to fix a new bug. In new bugs are manually triaged by an experts. Manual error triage is expensive in time cost and low in accuracy. To reduce the expensive cost of manual work in bug triage. An automatic bug triage approach, which uses text classification techniques. A bug triage is converted into a problem of text classification and is an organized solved with grown up text techniques. e.g., Naive Bayes. To increase the exactness of text classification technique for bug triage, some techniques are used, e.g., a tossing graph approach and a collaborative filtering approach, large-scale and low-quality bug data in bug repositories hide the techniques of automatic text classification.

How to minimize the bug data to save the time cost of developers and improve the quality of bug triage. Data reduction for bug triage achieves to make a scale is small and high-quality set of bug data by removing the bug reports and words. To combine two techniques of demo selection and feature selection to reduce the bug dimension and the word dimension. To evaluate the reduced bug data according to two methods: the scale of a data set and the accuracy of bug triage. Given an instance selection algorithm and a feature selection algorithm, the order of using these two algorithms may distress the results of bug triage. Based on the attributes from historical bug data sets.

1) To current the problem of data reduction for virus triages. This problem is to increase the data set of bug triage in two reasons a) to reduce the scales of the bug dimension and the word

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 2, March 2016

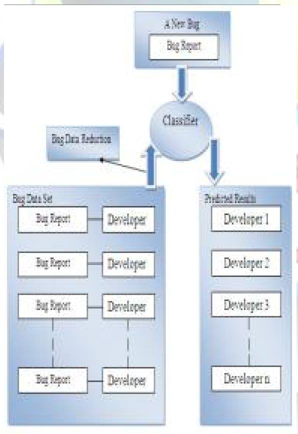
dimension b) to advance the reliability of bug triage.

2) The recommend a combination of approach to finding the problem of data reduction. This can be viewed as an application of instance mixture and feature mixture in bug repositories.

3) To make a binary classifier to expect the order of applying instance selection and feature selection. The order of applying instance selection and feature selection has not been investigated in related domains. To add new attributes get from bug data sets, prediction for order of reduction used. This causes an issue as different creators should not be assigned the same defect. Identify the bug reports are duplicate of others is typically done physically by a person called the triager. The triager would detect if a bug report is a copy; if it is, the triager would mark this report as a copy report and the first report as the master report.

BACKDROP AND DRIVE

BACKGROUND



Evaluate our approach on most bulky bug details datasets from big open source projects including Mozilla; A software raised area host more sub-projects such as Firefox browser. Thunderbird email client, Eclipse, a popular open source integrated development environment, and Open Office. Bug repositories are mostly used for managing software Bugs. a software bug is found, a reporter records this bug to the bug repository. The recorded bug is called a bug report. The techniques on text classification can be used to predict the developer for a new bug. In details, NAIV BAYES is atypical classifier is used in bug triage. New bug reports are treated as a test set to examine the results of the classification.

Motivation

The low-quality bugs build up in bug repositories with the growth in scale. Its bug tracking method is generally available to testers and even to all end users. Once a bug Manifests, people can submit a bug report depicting the detail of the bug.

WORKFLOW FOR RETRIEVING DUPLICATE BUG REPORT

In a duplicate report retrieval system, the fields review and report of both existing and new bug reports are preprocessed by usual information recovery methods. The type of each bucket is a master description, and its rate is a list of copy reports on the similar bug. Our approach consists of three aspects. First the main aim to develop the precision of textual similarity measures specially for bug reports. Second in order to improve the presentation of copy bug report recovery, in adding textual fields we struggle to take well again advantage of the other kinds of information available in bug reports. From the three pairs of copy reports. We note that duplicate reports are similar not only textually in review and description fields but also in the categorical fields such as product, component, and precedence.

DATA REDUCTION FOR BUG TRIAGE

Developers propose bug data reduction to improve the quality of data in bug repositories. To applied as a phase in data preparation of bug triage. To companioning the techniques of demo selection and features to remove certain bug reports and words. The creators first present how to apply instance selection and feature selection to bug data, i.e., data reduction for bug triage. The list the benefit of the data reduction in bug triage. Data reduction based on FS! IS

Input: training set T with n terms and m infection reports. Output: decreased data set T FI for bug triage.
1) Apply FS to n terms of T and estimate objective values for all the terms.
2) Apply IS to mI bug reports of T F.

A bug data set is changed into a text matrix with two dimensions. That is the bug dimension and the word dimension in bug triage. To combine of instance variety and characteristic selection to produce a compact bug data set. Instance selection and feature selection are mostly used techniques in data processing. Feature variety is to obtain a subset of applicable characteristics. To apply the mixture of occurrence variety and characteristic selection. Given an occurrence selection algorithm IS and a feature selection algorithm FS, to first applies FS and then IS. To decrease the bug data based on FS! IS. The output of bug data decrease of reduced data set.

Advantage of Data Reduction

The data reduction for bug triage has two goals, 1) reducing the data scale and 2) improving the accuracy and efficiency of bug triage. The contrast is aim to augment the dataset to build a preprocessing approach; it can be applied before an open bug triage approach.

Minimizing Data Scale

The creators are minimizing scales of data sets to save the time cost of developers. The aim of bug triage is to assign bug fixing. Once a developer is assigned to a new bug report, the developer. Here, use reducing duplicate and noisy bug reports to decreasing the number of selection. historical bugs. The labor and time cost of creators can be saved by reducing the number of bugs based on instance selection and feature selection.

REDUCTION ORDERS PREDICTION

The instance selection algorithm IS and a feature selection algorithm FS, FS! IS and IS! FS are the two orders for producing reducing techniques. A test is how to determine the order of Reduction techniques, i.e., how to choose one between FS! IS and IS! FS.

ORDER OF REDUCTION

Using the data reduction to each new bug data set, need to check the accuracy of both two orders. To reduce the time cost of manually checking both reduction orders. To consider predicting the decrease order for a new bug data set based on historical data sets. A error data set is mapped to a request and the related reduction order and is mapped to the label. Training a classifier once can get the reduction orders for all the new data sets without checking equally reduction orders. The problem of predicting reduction orders of using feature selection and instance selection has not been investigated in other application scenarios.

Bug Data Set and Attribute

To make a binary classifier to predict reduction orders. Before triaged new bugs such attributes can be extracted. It not provide adequate information for predicting correct developers. To remove the inventors, who have fixed less than 10 bugs. To conduct text classification, to extract the summary and the description of each bug report to denote the content of the bug. For a newly reported bug, the summary and the explanation are the most agent items, which are also used in physical bug triage.

DATA PREPARATION

Take the bug repositories of three big open source projects: OpenOffice, Mozilla and Eclipse. OpenOffice is an open source complement of Microsoft Office. Mozilla is a Community hosting several open source projects such as the famous web browser Mozilla fire box emails. Eclipse is an extensible multi-language software improvement Atmosphere written in Java. These three projects are varied in terms of purposes, users and implementation languages, thus help generalizing the ending of the experiments on them. By using text classification, to take out the summary and the explanation of each bug report to represent the content of the bug. As the input of classifiers, the rundown and the explanation are into the vector space model. To employ two steps to form the sound vector space, tokenization and stop word elimination. First, to tokenize the summary of bug reports into word vectors. Each word in a

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 2, March 2016

error report is associated with its Word frequency. Non-alphabetic words are removed to decrease the noisy sound. Then, remove the stop words; these are in high frequency and provide no needed information for bug. The list of stop lexis in our work is based on SMART information rescue system. Does not use the stemming technique has examined that is not help to error triage. Hence, the bug reports are transformed into vector space model.

Retrieval Function

To study the results of bug data reduction on bug repositories of two projects, Eclipse and Mozilla. Then evaluate results on five data sets and every data set is above 10,000 bug reports. To obtain over 10,000 fixed or duplicate bug information, every data set in Eclipse is gathered from continuous 20,000 bug reports in every bug set in Mozilla is collected from continuous 40,000 bug reports. To examine the results of data reduction, using four instance selection method and four feature selection methods IG, CH, SU, RF and Naïve Bayes. The results of data reduction for bug triage can be measured in two aspects, the scales of data sets and the quality of bug triage. The scales of data sets are configured as input parameters. The quality of error triage can be calculated with the accuracy of bug triage. It is defined as Accuracy ¼ # correctly assigned error reports in k candidate's# all error reports in the test set.

RATE OF DATA BUG REPORTS AND WORDS

Either case selection or characteristic selection method the number of instances or features should be determined to obtain the final scales of data sets. To investigate the changes of accuracy of bug triage by changeable the rate of selected bug reports in instance selection and the velocity of preferred words in feature selection. Enchanting two instance selection algorithms (ICF, LVQ) and two feature selection algorithms (IG and CH) For instance selection, ICF is a little better than LVQ. A good percentage of bug reports is 50 or 70 percent. In feature selection, CH always performs better than IG.It can find that 30 or 50 percent is a good percentage of words.

Results of Data Decrease for bug triage

Each instance selection method and each feature method algorithm based on one bug triage algorithm, Naive Bayes. To combine the best instance selection methods and the best feature selection methods to check the data reduction on three text-based bug triage method. Instance selection algorithms and four feature selection algorithms on four data sets. DROP or POP can achieve one best result. CH provides the best accuracy. IG and SU also complete good results. ICF, LVQ, and DROP give best results. In feature selection, CH also provides the best accuracy. To investigate the results of ICF and CH and to avoid the complete assessment on all the four instance mixture method and four feature mixture method. Feature mixture can increase the accuracy of bug triage over a data set and instance selection may reduce the correctness. On typical instance selection methods on classic data sets. The correctness reduced by instance selection is caused by the large number of creators in bug data sets. we sort developers by the number of their fixed bugs in sliding order is make by choosing the top developers.

DUPLICATE BUG REPORT

A bug report provides several functions. It is used to file imperfection, propose features and evidence maintenance responsibilities. A bug report consists of several fields. The fields in dissimilar projects may vary to some extent, but in common they are related. Table I lists the fields of importance to us in OpenOffice bug reports. Fields review and explanation are in normal language text, and to refer to them as textual characteristics, whereas the other fields try to find to illustrate the details from further view point and then refer to them as non-textual features or clear-cut feature.

DISCUSSION

The data decrease for bug triage to decrease the difficulties of data sets and to develop the feature of bug reports. By using illustration selection and feature selection to diminish noise and redundancy in bug data sets.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 2, March 2016

Here, not all the blare and redundancy are removed. Less than 50 percent of copy bug reports can be removed in data reduction. The correctness of bug triage is not good. This fact is foundation by the complication of bug triage.

CONCLUSIONS

Bug triage is a luxurious and time consuming step of software continuance in both labor cost and time cost. To combine feature selection and instance selection to reduce the scale of error data sets as well as improve the quality. To extract attributes of each bug data set and train a suitable model based on historical data sets. To provides a framework to leveraging on data processing to form reduced and high-quality bug data in software development and maintenance. Then to plan on improving the results of data diminution in bug triage to examine how to prepare a high quality bug data set and undertake a domain software task. For getting reduction orders, to plan to pay efforts to find out the possible relationship between the attributes of error data sets and the reduction orders.

REFERENCES

[1] 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. Bol_on-Canedo, N. S_anchez-Maro~no, 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] S. Kim, H. Zhang, R. Wu, and L. Gong, “Dealing with noise in defect prediction,” in Proc. 32nd ACM/IEEE Int. Conf. Softw. Eng., May 2010, pp. 481–490.

[12] A. Lamkanfi, S. Demeyer, E. Giger, and B. Goethals, “Predicting the severity of a reported bug,” in Proc. 7th IEEE Working Conf. Mining Softw. Repositories, May 2010, pp. 1–10.

[13] G. Lang, Q. Li, and L. Guo, “Discernibility matrix simplification with new attribute dependency functions for incomplete information systems,” Knowl. Inform. Syst., vol. 37, no. 3, pp. 611–638, 2013.

[14] D. Lo, J. Li, L. Wong, and S. C. Khoo, “Mining iterative generators and representative rules for software specification discovery,” IEEE Trans. Knowl. Data Eng., vol. 23, no. 2, pp. 282–296, Feb. 2011.

[15] Mozilla. (2014). [Online]. Available: http://mozilla.org/