Software Data Reduction Methodologies For Effective Bug Triage

MONIKA V S, Dr.M.Charles Arockiaraj

AMC ENGINEERING COLLEGE

ABSTRACT

Software companies experience asignificant decline of over 45% in revenue due to software issues. Bug triage plays a vital role in the bug-fixing process by assigning developers to new problems. To streamline and reduce manual effort in bug triage, automatic bug triage uses text categorization algorithms. Our research focuses on data reduction for bug triage, aiming to decrease data size while enhancing quality.

To achieve this, we combine instance and feature selection techniques to reduce the size of bug reports and individual words. Using variables from existing bug datasets, we develop a prediction model todetermine the optimal sequence forapplying instance and feature selection ona new bug dataset. To validate our approach, we conduct an empirical analysis on a dataset comprising 600,000 bug reports from Eclipse and Mozilla, two well-known open-source projects.

Our study's findings demonstrate that our proposed data reduction methodology effectively decreases data size whileimproving bug triage accuracy. This research offers a valuable technique for employing data processing methods to generate concise yet high-quality bug data in software development and maintenance

---

INTRODUCTION:

Software companies face a significant financial burden, allocating over 45% of their expenses to resolve software-related issues. Bug triage is a crucial step in the bug fixing process, involving precise assignment of developers to newly discovered bugs. Automated bug triage utilizes text categorization algorithms tooptimize resource allocation and reduce costs. This research focuses on data reduction in bug triage, enhancing bug data quality while reducing quantity.

We explore data reduction methods for bug reports and individual words, integrating instance selection and feature selection techniques. A prediction model is developed to determine the optimal sequence for applying these methods on a new bug dataset. Our approach is validated through empirical analysis on a dataset of 600,000 bug reports from Mozilla and Eclipse open-source projects.

Results demonstrate the effectiveness of our proposed data reduction methodology, reducing data volume while improving bug triage accuracy. Leveraging advanced data processing techniques, our research offers a practical solution for concise and high-quality bug data generation, benefiting software development and maintenance processes.

1. EXISTING SYSTEM:

• Hong et al. develop a social platform for software developers, enabling them toexplore their contributions to the Mozilla project using bug-related information. Thisplatform enhances understanding of the developer community and project progress.

• Xuan et al. analyze open-source software bug repositories, associating issue priority with specific developers. This analysis provides insights into developer prioritization and streamlines software maintenance responsibilities.

• Zimmermann et al. conducted... conduct surveys among developers and customers participating in three open-source projects to evaluate problem-related data quality. Survey results are used to identify crucial characteristics of effective bug reports andtrain a classifier for further improvement identification.

2. PROPOSED SYSTEM:

This article explores data reduction in bug triage to minimize the volume of bug- related data. Our main objective is to achieve cost savings, improved data quality, and a more efficient bug triageworkflow.

Data reduction in bug triage involves condensing bug data while maintaining its quality. This includes tasks like eliminating redundant or irrelevant problem reports and phrases. We focus on removing duplicate bug reports, merging similar reports, and filtering out non- informative terms.

MODULES:

- Instance Selection
- Data Reduction
- Graph Module

DESCRIPTION OF MODULES:

Instance Selection:

Instance selection and feature selectionare common techniques in data processing with diverse applications. Instance selection involves extracting relevant instances from a dataset, such as bug reports from bug data,

while feature extraction aims to extract relevantfeatures, such as specific words, for a particular dataset used in a specific application. The purpose of these techniques is to obtain subsets of important instances and features within a dataset. In our study, we use a hybrid approach that combines instance selection and feature selection methods customized for bug data.

Data Reduction:

Our research focuses on data reductionfor bug triage, with the goals of reducing labor costs and improving triage quality.

Addressing data sparsity: We aim to minimize gaps between data points, condensing the dataset for better efficiency and manageability.

Enhancing triage reliability: Our objective is to improve bug assignment accuracy and effectiveness, enhancing the dependability of the triage process.

Our approach involves preprocessing the dataset, seamlessly integrating with existing bug triage procedures. Unlikeprevious studies that focused primarily onmodeling bug reports as text, our approach complements established triage methods. The following provides more details on the objectives of data reduction.

Graph Module

This section consists of fourcomponents:

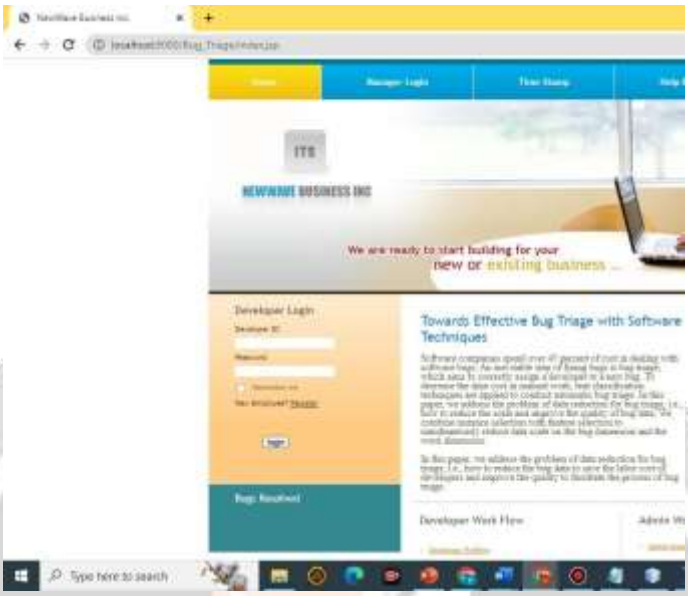
1. Initial component: Provides an overview of unresolved issues awaiting assignment to developers. The administrator receives comprehensive status information to identify pending bugs.

2. Defects without assigneddevelopers: Displays the number of bugs without assigned developers.It helps the administrator track bugsawaiting responsibility.

3. Resolved bugs by developers:Indicates the number of bugssuccessfully addressed by developers. It provides extensive
bug information to identify resolvedissues.

4. Pending bugs for developers: Displays the count of bugs yet to be addressed by developers. Theadministrator receives detailed information about unresolved issues.

RESULT:


NEWWAVE BUSINESS INC
We are ready to start building for your
new or existing business ...
Developer Login
Towards Effective Bug Triage with Software Techniques
Developer Work Flow
Type here to search

Conclusion:

Software maintenance entails substantial costs in terms of labor and time, particularly during bug triage. By extracting attributes from bug data sets and training a prediction model using historical data, we determine the optimal sequence for applying instance selection and feature selection to new bug data sets. This serves as a reference point for future data sets. Our empirical study focuses on data reduction techniques for bug triage, utilizing repositories from Mozilla and Eclipse open-source projects. The study demonstrates how data processing approaches can generate concise bug data sets without compromising quality, benefiting software development and maintenance processes.

As part of our ongoing research, we aim to improve data reduction outcomes in bug triage. Our objective is to explore effective methods for preparing high-quality domain-specific bug data collections. Additionally, This effort will contribute to further advancements in our ability to predict optimal reduction sequences.

REFERENCE:

1. "Determining the Responsible Developer for Bug Fixing" was published by J. Anvik, L. Hiew, and G. C. Murphy in the Proceedings of the 28th International Conference on Software Engineering in May 2006, on pages 361–370.

2. S. Artzi, A. Kiezun, J. Dolby, F. Tip, D. Dig, A. Paradkar, and M. D. Ernst, "Identifying Bugs in Web Applications using Dynamic Test Generation and Explicit-State Model Checking," IEEE Software, vol. 36, no. 4, pages 474–494, July/August 2010.

3. J. Anvik and G. C. Murphy authored a paper titled "Effort Reduction in Bug Report Triage: Recommendations for Development-Oriented Decisions." It can be found in Volume 20, Issue 3 of the ACM Transactions on Software Engineering Methodology, published in August 2011.

4. C. C. Aggarwal and P. Zhao, "Graphical Models for Text Processing," in Knowledge and Information Systems, vol. 36, no. 1, pages 1–21, 2013 Bugzilla, (2014). [Online]. Available: http://bugzilla.org/

5. K. Balog, L. Azzopardi, and M. de Rijke, "Formal Approaches for Identifying Experts in Enterprise Corpora," in Proceedings of the 29th Annual International Conference of the Association for Computing Machinery Special Interest Group on Research, Development, and Information Retrieval, August 2006, pages 43–50.

6. P. S. Bishnu and V. Bhattacherjee, "Software Fault Prediction using Quad Tree-Based K-means Clustering Algorithm," published in IEEE Transactions on Knowledge and Data Engineering, vol. 24, no. 6, June 2012, pages 1146–1150.

7. H. Brighton and C. Mellish, "Advancements in Instance Selection for Instance-Based Learning Algorithms," Data Mining and Knowledge Discovery, vol. 6, no. 2, pages 153–172, April