

Test Logs Segregation and Analysis Automation

Sharath Krishna Karanam¹, Soumya A²

¹B.E Student, Department of Computer Science Engineering, RV College of Engineering, Bengaluru

²Associate Professor, Department of Computer Science Engineering, RV College of Engineering, Bengaluru

Abstract - Software corporations are working with enormous number of test failures daily. Test triage fixes failures by assigning triage engineers to a newly encountered failure. Software corporations spend hefty amount in fixing these test failures. Every product is tested thoroughly before releasing it the customers. As of today, test triaging is done manually. This ends up in a lot of time consumption and manual effort. Automation of this process reduces the time consumed drastically. Test Failures need to be fixed in timely manner as it directly aids the release of the product. The current paper addresses review of existing approaches of fixing Test Failures and also proposes a method for the same to handle redundancy in test repository. In order to minimize such effort, data comparison logic can used to distinguish between old and new test failure which saves time and eliminates redundant data.

Key Words: Automation, Test Failures, Triage, Support Vector Machine (SVM), Vector Machine Content-Based Recommendation (CBR), Content-Boosted Collaborative Filtering (CBCF)

1. INTRODUCTION

Software corporations invest more time triaging the test failures of the product. Industry project maintains a database which holds data of test failures. Every test failure in database contains a detailed data called as log data. The database holds textual data of the failure which is very difficult for new developer to interpret. Every day 6-7 hours are spent on triaging and fixing the test failure, thus automation for the same saves time and manual effort. The Test Triage meeting ought to be encouraged by Quality Assurance (QA) lead and this gathering ought to be led in testing period of the Systems Development Life Cycle (SDLC). The recurrence of the gathering is changing from undertakings to activities and it is to be chosen dependent on what number of failures are logged by the analyzers. The point is to make a move on generally significant or basic test failures first and not thought a lot to fix non-significant test failures. Triage in programming improvement requires three gatherings be available. Test encourages a failure triage with a test failure following framework and peruses the depiction of the inability to everybody, item for the most part poses some explaining inquiries.

1.1 Comparison Logic

The Levenshtein separation is a measurement to gauge how comparative two groupings of words are. As it were, it

quantifies the base number of alters that has to be done to change a single word arrangement into the other. These alters can be inclusions, cancellations or substitutions. This strategy was named after Vladimir Levenshtein, who initially thought about it in 1965. Note that the lines on the base above compare to an expulsion, an addition, and a substitution in a specific order. It is conceivable to register the Levenshtein likeness proportion dependent on the Levenshtein separation. This can be done using the Equation (1):

$$\text{lev}_{a,b}(i, j) = \begin{cases} \max(i, j) & \text{if } \min(i, j) = 0, \\ \min \begin{cases} \text{lev}_{a,b}(i-1, j) + 1 \\ \text{lev}_{a,b}(i, j-1) + 1 \\ \text{lev}_{a,b}(i-1, j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases} \quad (1)$$

where a, b are strings,

i, j are lengths of a, b respectively.

This is implemented using fuzzywuzzy library in python which is used to get text similarity. To get increasingly precise closeness measure cosine similarity is utilized. The cosine similarity between two vectors (or two records on the Vector Space) is a measure that figures the cosine of the point between them. This measurement is an estimation of direction and not size, it very well may be viewed as an examination between records on a controlled space as just the size of each word check of each report is not thought of, yet the edge between the archives. Cosine similarity is represented as in Equation (2):

$$\text{Cos } \alpha = \frac{a \cdot b}{|a| \cdot |b|} \quad (2)$$

Where α is the angle between vectors 'a' and 'b'

The log files text similarity range between 0 and 1.

2. LITERATURE REVIEW

Existing methods treats Test triage as an issue of suggestion and proposes an answer which is an occasion of Content Based Recommendation (CBR). Be that as it may, CBR experience the ill effects of over-specialization for example the test failures that every engineer has settled before. In this way, Park et al. proposed another Test failure Triage Technique which right off the bat changes over the test failure triage into an improvement issue; enhancing exactness and cost. At that point they embraced a Content Boosted Collaborative Filtering (CBCF) which joins a current CBR with a Collaborative Filtering Recommender (CF). Creators call attention to that to accomplish these two objectives, a key test is meager condition. To address this test,

they built up a theme model to diminish the scantiness and improve the nature of CBCF [1].

With the expanding attention to the issues identified with the administration of test failures, numerous scientists apply the current information mining and content handling methods to test failure storehouses as of late. Cubranic et al. first acquainted content grouping system with test failure task in 2004 and effectively anticipate 30% test failure resolver in Eclipse test failure store utilizing Naive Bayes [1]. Anvick et al. extended the work and did a correlation among the consequences of utilizing choice tree, Naive Bayes, and Support Vector Machine [2].

2.1 Existing Automation Tool

The CERT Triage Tools can be utilized to help programming dealers and investigators in recognizing the effect of imperfections found through procedures, for example, fluff testing and organizing their coordination in the product advancement process. The CERT Triage Tools incorporate a GNU Debugger (GDB) augmentation called "exploitable" that orders Linux application test failures by seriousness and a wrapper content for clump execution. CERT Triage Tools venture has been progressed to the GDB 'exploitable' module venture on GitHub. The CERT Triage Tools were created to fill needs on the Linux stage. It is good with 32-bit or 62-bit Linux stage. It helps in gathering, examining, and approving the rise vulnerabilities to normal registering stages [3].

2.2 Test Triage Recommenders

Author introduced a Machine Learning (ML) approach to manage triage helping recommenders [3]. They notice it as ML Triage. The objective of this is to decrease the human contribution in triage. Right now process, initially, from a task's issue following framework reports are consequently chosen. At that point, from these chose reports, highlights are gathered and reports with same highlights are assembled under a name. The name demonstrates the class to which the highlights have a place. Recommender is made for explicit advancement situated choice. When recommender creates a forecast for new test failure report at that point time highlights are separated and are taken care of to the recommender.

2.3 Categorization Methods

Author tended to the issue of information reduction for test failure triage [4]. The test failure triage attempts to foresee the engineers who can fix the test failures trailed by current task to evacuate unfixed test failure reports. They measured just the test failure reports those are fixed and repetitive dependent on the test failure report status in the database. Guileless Bayes system is utilized for content characterization. The occurrence determination with include choice is consolidated to diminish information scale for the sake of test failure measurement and the word measurement. To gain the request for applying event assurance and feature decision, from the recorded test failure informational collections the extraction of characteristics is thought of. J. Anvick et al. introduced a semi-computerized approach for the task of reports to a designer for extra taking care of [5]. This approach depends on a managed AI calculation which is applied to data available in the test failure archive. In

Artificial Intelligence (AI) the reports are called occasions and trait of example are called highlights. The classifier creates the directed AI procedure that offers few engineers appropriate to determine the new report on appearance. J. Xuanett al. talked about a semi-managed content characterization approach for test failure designer task, to dodge the lack of marked test failure reports saw in other actualized directed methodologies [6]. They consolidated Naive Bayes classifier and Expectation Maximization for better results.

2.4 Automatic test failure assignment

There are numerous procedures for test failure triage like: Machine learning Incremental getting the hang of, Tossing Graph. These procedures are useful for triaging and diminishing hurling way; however, their precision diminishes by different issues like obsolete preparing sets, inert designers and so forth. At the point when a test failure report has been reassigned to different engineers for fixing the test failure that recently allotted designer cannot fix, this procedure. G. Jeoung et al. concentrated on 445,000 test failure reports just as their general exercises [7]. For improving the test failure task process and decrease un-required hurling steps, they utilized hurling chart model that utilized existing hurling history. Author enhanced triaging and diminished hurling way lengths by utilizing a few systems [8]. Creators' suggest that the current frameworks AI systems are ineffectual for huge venture. They suggested three unique augmentations to existing methods. Authors guarantee to accomplish higher forecast exactness utilizing better element vectors. They utilized a total arrangement of AI instruments just as a probabilistic chart-based model (test failure hurling diagrams) by which they guarantee high precision forecasts, and establishment for AI based test failure task. They utilized procedure like Choosing viable classifiers and highlights, Incremental learning, Multi included hurling diagrams to accomplish their objective.

2.5 Classification Techniques

Naive Bayes is part of Bayesian Decision Theory. The name "gullible" originates from the way that it marks a credulous suspicion that each pair of highlights being arranged is free of one another. In Bayes' Theorem, the likelihood of an occasion happening is determined given the likelihood of additional happened occasion as in mathematical equation (3):

$$P(H | X) = P(X | H) P(H) P(X) \quad (3)$$

Where $P(H | X)$ is the probability of hypothesis given that is true.

$P(H)$ is probability of hypothesis .

$P(X)$ is probability of event.

$P(X | H)$ is the probability of event given that probability of is true.

Support Vector Machine (SVM) is a directed learning classifier which executes the idea of isolating hyperplane to build outskirts among classifications and afterward utilizes them to order new information. For instance, in two-dimensional space, the hyperplane is a line with each side has a place with a class. Despite the fact that the first SVM is a

straight classifier, utilizing portion which maps information into high dimensional highlights space can assist it with performing different types of tests also.

3. CONCLUSIONS AND FUTURE DIRECTIONS

In general, automating test failure fix helps in scanning through the test logs without the manual intervention minimizing the developer's effort. These techniques are mainly used by the triage engineers to get a more readable format than the traditional log files which take lots of effort and time to understand. New developers who are unaware of test architecture have problem understanding. Developers can access only the failure part and work on it without much effort.

Survey shows most of the papers address about accuracy and very less work has been reported on handling redundancy in fixing Test Failures. Redundancy can be handled using comparison logic, which saves memory and time to fix the failure by maintaining an additional library for test failures. A website with user friendly UI can be designed which enables easy access to the details of failure tests where in developer can query according to their requirement. This proposed approach segregates and analyze the log data which gives more accurate and readable results. All tests usually run on non-working hours and also performed sequentially. In future test run time can be optimized by using parallel approaches and also ensure the correctness of the test run.

REFERENCES

- [1] Trupti S. Gadge, Nikhil Mangrulkar 'Approaches for Automated Bug Triaging' International Conference on Innovative Mechanisms for Industry Applications (ICIMIA 2017) 978-1-5090-5960-7/17/\$31.00 ©2017 IEEE
- [2] Yuan Gao, James B Mann, Srinivasan Chandrasekar, Ronglei Suna, Jürgen Leopold, Modelling of Tool Temperature in Modulation-assisted Machining, 16th CIRP Conference on Modelling of Machining Operations, 58, 204-209, 2017
- [3] P. Bholowalia and A. Kumar, "Ebk-means: A clustering technique based on elbow method and k-means in wsn," International Journal of Computer Applications - Volume 105, pp. 18-19, 2014.
- [4] A. K. Farahat, A. Ghodsi, M. S. Kamel, "Efficient greedy features selection for unsupervised learning," Knowl. Inform. Syst., vol. 35, no. 2, pp. 285-310, May 2013.
- [5] Xuan-Vinh Nguyen, Phu-Khoa Nguyen, Vu Nguyen 'Clustering Automation Test Faults' 978-1-7281-3003-3/19/\$31.00 ©2019 IEEE
- [6] V'aclav Ded'ik, Bruno Rossi 'Automated Test failure Triaging in an Industrial Context' 42th Euromicro Conference on Software Engineering and Advanced Applications, 2016
- [7] C. C. Aggarwal and P. Zhao, "Towards graphical models for text processing," Knowl. Inform. Syst., vol. 36, no. 1, pp. 1-21, 2013.
- [8] N. DiGiuseppe and J. A. Jones, "Concept-based failure clustering," Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering, 2012.
- [9] Alaasam Jifeng Xuan, He Jiang, Yan Hu, Zhilei Ren, Weiqin Zou, Zhongxuan Luo, and Xindong Wu, "Towards Effective Test failure Triage with Software Data Reduction Techniques", IEEE transactions on knowledge and data engineering, vol. 27, no. 1, January 2015
- [10] J. Xuan, H. Jiang, Z. Ren, J. Yan, and Z. Luo, Automatic test failure triage using semi-supervised text classification," in Proc. 22nd Int. Conf. Softw. Eng. Knowl. Eng., Jul., pp. 209-214, 2010
- [11] T. Zimmermann, N. Nagappan, P. J. Guo, and B. Murphy, "Characterizing and predicting which test failures get reopened," in Proc. 34th Int. Conf. Softw. Eng., Jun. 2012
- [12] S. Romano, N. Vinh, J. Bailey, and K. Verspoor, "Adjusting for chance clustering comparison measures," The Journal of Machine Learning Research 17, pp. 1-32, 2016.
- [13] P. S. Bishnu and V. Bhattacharjee, "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.
- [14] J. Hirschberg and A. Rosenberg, "V-measure: A conditional entropy based external cluster evaluation measure," EMNLP-CoNLL, vol. 7, pp. 410-420, 2007
- [15] I. H. Witten and E. Frank. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, San Francisco, 2nd edition, 2005.