READING ASSIGNMENT I

A Topic-Based Approach for Narrowing the Search Space of Buggy Files from a Bug Report

Anh Tuan Nguyen, Tung Thanh Nyugen, Jafar Al-Kofani, Hung Viet Nyugen, Tien N. Nyugen

IMPORTANT KEYWORDS

ii1. Topic Based Search Approach

Search engine Personalization based on topical ontology to identify user's interest. The topics include general topics which are of interest to users and then a classification technique is used to classify the search queries. User's preferences is identified which is then used to personalize search results.

ii2. Defect Localization

This is process to analyze the bug(s) and then search through the program code base to find the potential buggy/defective files.

ii3. Generative Machine Learning Model

Generative model classifies data and generates labels based on joint probability distribution i.e. P(x,y). Analytic models like Latent Dirichlet Model, Naive Bayes use generative Machine Learning model.

ii4. Empirical Evaluation

The appraisal of a theory by observation in experiments.

BRIEF NOTES

iii1. Motivational Statements

Defect localization consumes a lot of time in process of software development. If this time spent can be reduced, it can improve the bug fixing time. Topic based search approach as proved to work in web search personalization can be used to narrow buggy source code files or reduce search space considerably.

iii2. Hypotheses

Narrowing down the search space of Buggy Files(with an efficiency of about 45%) has a great potential to decrease the bug file localization time in software industry by a considerable amount of time. This information can be used to concentrate on the potential buggy area(s) which be targeted in regression tests, thus helping to deliver a more comprehensive software product.

iii3. Related Work

Lurkins et al.[1] applied Latent Dirichlet Analysis on bug reports and and source files to narrow down search space, however they did not take into account the repetition of common terms in code. Premraj et al.[2] address bug localization by combining textual features in the document with the list of frequently buggy locations. Ostrand et al. [3] developed a model based on code and modification history of code to predict expected number of faults in them in next release.

iii4. Commentary

Latent Dirichlet Analysis (LDA) is used to model relationship between bug report and buggy source files. There are two components called S-component which is LDA for a source file and B-Component which is extended LDA model which is modeled as a document by its own topic distribution and also topic distribution of source files.

IMPROVEMENTS

iv1. Analysis could have been done on the successful match results to narrow down if match was due to the nature in which the bug was reported. Suggestion(s) to include keywords in bug report would have improved the performance of the model and hence made the paper more relevant.

iv2. Author has not mentioned the ratio of bug reports and the phase of testing. Unit test bug report which are filed by developer who has written the code is expected to have lot more details related to source code whereas report filed by a black box tester may not have enough data. Analyzing the data for different phases would have given much accurate results.

iv3. The source code file and bug reports which have been studied are specific to a particular company. Running the model on a wider base on data coming from different fields would have given a lot more accurate result of the accuracy of the model.

REFERENCES

1. S. K. Lukins, N. A. Kraft, and L. H. Etzkorn. Bug localization using latent dirichlet allocation. J. of Inf. Softw. Technol., 52(9):972–990, 2010.

2. R. Premraj, I.-X. Chen, H. Jaygarl, T.N. Nguyen, T. Zimmermann, S. Kim, and A. Zeller. Where should I fix this bug? (bugtalks.wikispaces.com), 2008.

3. T. J. Ostrand, E. J. Weyuker, R. Bell. Predicting the location and number of faults in large software systems. IEEE TSE, 31(4):340–355, 2005.