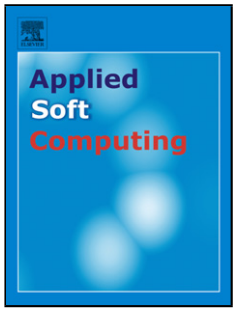

Title: Comparing Learning to Rank Techniques in Hybrid Bug Localization

Authors: Zhendong Shi, Jacky Keung, Kwabena Ebo Bennin, Xingjun Zhang

PII: S1568-4946(17)30662-2
DOI: https://doi.org/10.1016/j.asoc.2017.10.048
Reference: ASOC 4543

To appear in: *Applied Soft Computing*

Received date: 9-5-2017
Revised date: 21-10-2017
Accepted date: 29-10-2017

Please cite this article as: Zhendong Shi, Jacky Keung, Kwabena Ebo Bennin, Xingjun Zhang, Comparing Learning to Rank Techniques in Hybrid Bug Localization, Applied Soft Computing Journal https://doi.org/10.1016/j.asoc.2017.10.048

Comparing Learning to Rank Techniques in Hybrid Bug Localization

Zhendong Shi[1,2], Jacky Keung[2], Kwabena Ebo Bennin[2], Xingjun Zhang[1]

[1]Department of Computer Science and Technology
Xi'an Jiaotong University
Xi'an, China
zdshi0@stu.xjtu.edu.cn; xjzhang@mail.xjtu.edu.cn

[2]Department of Computer Science
City University of Hong Kong
Hong Kong, China
jacky.keung@cityu.edu.hk;
kebennin2-c@my.cityu.edu.hk

HIGHLIGHTS

- A summary of recent Information Retrieval based hybrid bug localization techniques.
- 8 Learning to Rank techniques are tested for bug localization.
- Coordinate ascent is the best performing LtR technique in selected attributes.

ABSTRACT

Bug localization is a software development and maintenance activity that aims to find relevant source code entities to be modified so that a specific bug can be fixed on the basis of the given bug report. Information retrieval (IR) techniques have been widely used to locate bugs in recent decades. These techniques mainly use the IR similarity between the bug report and source code entities. In addition to IR similarity, features that are extracted from version history, source code structure, dynamic analysis, and other resources are found to be beneficial for bug localization. The approaches utilizing extra features in IR-based bug localization are called hybrid bug localization. We conduct a short survey of the hybrid bug localization methods that use additional features in addition to IR similarity. We also use Learning to Rank (LtR) techniques to combine the beneficial features to improve bug localization. Learning to Rank is the application of machine learning in the ranking models for information retrieval. We compared eight LtR techniques in bug localization, and the experimental results show that coordinate ascent algorithms without normalization is a suitable LtR technique in bug localization for selected attributes, and it outperforms two state-of-the-art localization approaches for two large projects, Eclipse and SWT.

Keywords

Bug Localization; Information Retrieval; Learn to Rank; Fault Localization

1. INTRODUCTION

A software project in development or maintenance phases most likely contains bugs. Bug localization is the process of using automatic or semi-automatic techniques to find the buggy locations in a source code. Bug localization has elicited considerable attention in recent decades. As developers usually name the classes/methods/variables after their functionalities and write comments with relevant terms, the submitted bug reports typically include terms related to buggy functionalities. The bug reports are therefore likely to have similar textual terms with relevant source-code entities. Information retrieval (IR), the main technique used in bug localization, refers to the process of finding documents that satisfy an information need from a large collection of documents [33]. Many IR-based bug localization techniques have been studied in recent years [32, 43, 45, 50, 52, 56, 58].

In addition to mainly depending on IR similarities, many hybrid approaches that use additional features from source code structure [2, 47, 50], dynamic analysis [20, 30, 42], version history [51, 58], and others have been proposed in the literature. These hybrid approaches use additional features in favor of bug localization, and therefore perform significantly better than the approaches that use only IR similarities. A survey of these hybrid methods and their corresponding features is presented in this paper. However, many features are linearly combined with preset weight values [45, 52, 58, 59, 67], thereby indicating that the linear weight values are not learned from the data.

Combining the found beneficial features in a suitable manner may improve the performance of bug localization. Learning to Rank (LtR) is a set of machine learning-based ranking algorithms in the information retrieval domain. It has been widely used in web searching area in the