IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 3 Issue 8, August 2016
ISSN (Online) 2348 – 7968 | Impact Factor (2015) - 4.332
www.ijiset.com

An Efficient Code Clone Detection Model on web Applications: Review

Muneer Ahmad

RN engineering&management

college Department of cse

Mudasir ahmad mutto

Assistant professor SSM college of engineering&Technology

Abstract: **Code clones have been studied for long, and there is strong evidence that they are a major source of software faults. The copying of code has been studied within software engineering mostly in the area of clone analysis. Software clones are regions of source code which are highly similar; these regions of similarity are called clones, clone classes, or clone pairs In this paper a hybrid approach using metric based technique with the combination of text based technique for detection and reporting of clones is proposed. The Proposed work is divided into two stages selection of potential clones and comparing of potential clones using textual comparison. The proposed technique detects exact clones on the basis of metric match and then by text match.**

Introduction:

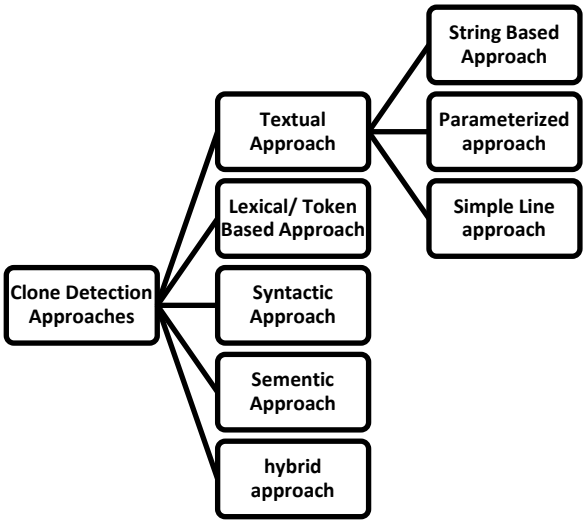
Code clones have been studied for long, and there is strong evidence that they are a major source of software faults. The copying of code has been studied within software engineering mostly in the area of clone analysis. Software clones are regions of source code which are highly similar; these regions of similarity are called clones, clone classes, or clone pairs. There are several reasons why two regions of code may be similar, the majority of the clone analysis literature attributes cloning activity to the intentional copying and duplication of code by programmers; clones may also be attributable to automatically generated code, or the constraints imposed by the use of a particular framework or library. Software clones Recently, various clone detection techniques have been subject to empirical comparisons to compare how well they perform.

Cloning works at the cost of increasing lines of code without adding to overall productivity. It results to excessive maintenance cost. Along with these negative impacts Due to unnecessary increase in complexity and length it becomes more difficult to edit the code hence, leading to increased human errors, high maintenance cost, forgotten or overlooked codes, and increased size of the code. Also along with the duplication of code it also duplicates the errors thus increasing the errors in the code file and decreasing its efficiency.

Identifying code clones serves many purposes, including studying code evolution, performing plagiarism detection, enabling refactoring such as procedure ex-traction, and performing defect tracking and repair. Most previous work on code-clone detection has focused on finding identical clones, or clones that could be made identical via consistent transformations of identifiers and literals. However, code segments that are similar but not identical occur often in practice, and finding such non-identical clones can be as important as finding identical code segments

The basic classification of the clones can be summed up in two categories *Textual clones* and *Functional clone.*

There are number of developers that have studied the techniques for the detection of the clones and there are number of approaches for the purpose of clone detection in any program. Few of them being Textual approach, Token-Based approach, Syntactic approach and Semantic approach.



Clone detection process:

Clone detection process involves several steps that are:

- Pre-processing
- Transformation
- Extraction
- Normalization
- Match-detection
- Formatting
- Post-processing
- Aggregation

Literature Review:

PriyankaBatta[1] Software Clone detection helps in detecting duplicate code from applications. Cloning creates problem when a bug is found in one code segment that was copied and pasted at several locations earlier. The objective of this study is to analyze the working of hybrid clone detection technique that design and analyze a hybrid technique for detecting software clone in an application. A model will be designed to automate the concept of clone detection.

Ali Selahmat and Norfaradilla Wahid [2] as the number of web pages increases across time number of clones among source code also increases. Aim is to be familiar with ontology mapping technique to solve the clone detection between files of different systems.

Florian Deissenboeck et al[3] Cloned code is considered harmful for two reasons: (1) multiple,possibly unnecessary, duplicates of code increase maintenancecosts and, (2) inconsistent changes to cloned code can create faults and, hence, lead to incorrect program behavior. Based on an industrial case study undertaken with the BMW Group, this paper details on these challenges and presents solutions to the most pressing ones, namely scalability and relevance of the results. Moreover, we present tool support that eases the evaluation of detection results and thereby helps to make clone detection a standard technique in modelbased quality assurance.

Chanchal K. Roy, et al [4] They provide a qualitative comparison and evaluation of the current state-of-the-art in clone detection techniques and tools, and organize the large amount of information into a coherent conceptual framework.

Shinji Kawaguchi,et al [5] code clones decrease the maintainability and reliability of software programs, thus it is being regarded as one of the major factors to increase development/maintenance cost.

Hoan Anh Nguyen, et al [6] Structure-oriented approaches in clone detection have become Popular in both code-based and model-based clone detection. However, existing methods for capturing structural information in software artefacts are either too computationally expensive to be efficient or too light-weight to be accurate in clone detection.

Marat Akhin and Vladimir Itsykson [7] This paper is a brief overview of current cutting edge in clone recognition. In the first place, they highlighted primary wellsprings of code cloning. Second, they diagrammed significant negative impacts that clones have on programming improvement. The most genuine downside copied code have on programming support is expanding the expense of adjustments - any alteration that progressions cloned code must be proliferated to each clone example in the project. Third, they surveyed existing clone discovery procedures. Arrangement in light of utilized source code representation model is given in this work. We likewise depict and dissect some solid case of clone location methods highlighting fundamental particular components and issues that are available in pragmatic clone recognition. At last, they called attention to some open issues in the zone of clone discovery. Right now addresses like "What is a code clone?", "Would we be able to anticipate the effect clones have on programming quality" and "By what method would we be able to increment both clone discovery accuracy and review in the meantime? " stay open to further re- - look. We list the most essential inquiries in advanced clone location and disclose why they keep on remaining unanswered in spite of all the advancement in clone identification research.

Kwantae Cho Minho Jo ; Taekyoung Kwon ; Hsiao-Hwa Chen ; Dong Hoon Lee[16] it is basic to distinguish clone hubs speedily to minimize their harm to WSNs. As of late, different clone discovery plans were proposed for WSNs, considering distinctive sorts of system setups, for example, gadget sorts and sending techniques. With a specific end goal to pick a powerful clone discovery plan for a given sensor organize, the choice criteria assume a vital part. In this paper, we first examine the choice criteria of clone location plans as to gadget sorts, recognition systems, organization techniques, and identification ranges. We then characterize the current plans as indicated by the proposed criteria. Reproduction trials are directed to think about their exhibitions. It is inferred that it is helpful to use the matrix arrangement learning for static sensor organizes; the plan utilizing the lattice sending information can spare vitality by up to 94.44% in tantamount execution (particularly as far as clone location proportion and the fulfillment time), when contrasted with others. Then again, for versatile sensor organizes, no current methodology works effectively in diminishing discovery mistake rate.

Conclusion: After this and more of the literature review the research work was decided on the hybrid approach for detecting the clone and it was noted that they can be potential improved when considered on the basis of the text and metric match techniques.

References:

[1] BattaPriyanka," Hybrid technique for software code clone detection" International Journal of Computer and technology, Volume 2, Issue 2, April 2012

[2] Selamatali&wahidnorfaradilla,"code clone detection using string based matching technique" International symposium on empirical software engineering" volume 2, Issue 4 April, 2005

[3] Deissenboeck Florian, Hummel Benjamin JuergensElmar, Pfaehler Michael," Model clone detection in Practice" ICSE Vol. 2, Issue 3, September 2008

[4] Roy K Chanchal,Cordy R James, KoschkeRainer " Comparison and evaluation of code detection techniques and tools" volume 2 issue 24 Febuary, 2009

[5] Kawaguchi Shinji, YamashinayTakanobu, UwanozHidetake, FushidaKyhohei, Kamei Yasutaka , NaguraMasatakaandlidaHajimu "Shinobi: A Tool for AutomaticCode Clone Detection in the IDE" ISSN : 2229-3345 Vol. 4 No. 06 Jun 2013

[6] Nguyen AnhHoan, Nguyen ThanhTung,Pham . H Nam, and Nguyen N Tien"Accurate and

Efficient Structural Characteristic Feature Extraction for Clone DetectionVolume 2 issue 8,September 2010.

[7] PuriAmit, "software code clone detection model" Vol. No.1, Issue 3, October 2012

[8] Chanchal K. Roy and James R. Cordy, "An Empirical Study of Function Clones in Open Source Software", 1095-1350/08 $25.00 © 2008 IEEE

[9] Mark Gabel Lingxiao Jiang Zhendong Su, "Scalable Detection of Semantic Clones", ICSE'08, May 10–18, 2008, Leipzig, Germany. Copyright 2008 ACM

[10] Chanchal K. Roy, "Detection and Analysis of Near-Miss Software Clones", 978-1-4244-4828-9/09/$25.00 2009 IEEE

[11] Yoshiki Higo, and Shinji Kusumoto, "Enhancing Quality of Code Clone Detection with Program Dependency Graph", 2009 IEEE

[12] ImanKeivanloo, JuergenRilling, Philippe Charland, "SeClone - A Hybrid Approach to Internet-scale Real-time Code Clone Search", 1063-6897/11 $26.00 © 2011 IEEE

[13] Randy Smith and Susan Harwtiz-detecting and meausingsimilaties in code clones.

[14] Chanchal K. Roy [,a], James R. Cordy[a], Rainer Koschke[b]-Comparison and Evaluation of Code Clone Detection Techniquesand Tools: A Qualitative Approach

[15] K. Kontogiannis, R. DeMori, E. Merlo, M. Galler, and M. Bernstein, Pattern Matching for Clone and Concept Detection, Journal of Automated Software Engineering, 3(1-2):77-108 (1996).

[16] Kwantae Cho Minho Jo ; Taekyoung Kwon ; Hsiao-Hwa Chen ; Dong Hoon Lee: Classification and Experimental Analysis for Clone Detection Approaches in Wireless Sensor Networks