Reading Assignment 7

Bhanu Anand, Esha Sharma, Vinay Todalge

i. Reference to the paper:

Xin Xia, David Lo, Xinyu Wang, and Bo Zhou
"Accurate Developer Recommendation for Bug Resolution" in Reverse Engineering (WCRE), 2013 - 20th Working Conference on 14-17 Oct. 2013.

Relation to First Paper:

This paper refers to our first paper for stating there has been numerous studies has been done for proposing automatically detection of duplicate bugs.

ii. Important Key-words:

1. Developer Recommendation for Bug Resolution

   This is relatively new approach of automatically processing new bugs and recommending a list of developers who are likely to resolve given bug.

2. Bug resolution

   Bug resolution refers to the activity that developers perform to diagnose, fix, test, and document bugs during software development and maintenance. It is a collaborative activity among developers who contribute

3. BR based Analysis

   In this analysis, for a given bug report, corresponding duplicate bug reports were discovered and appropriate developers are found based on developers of past duplicates found for given bug.

4. D based Analysis

   In this analysis, affinity/distance of a developer towards given bug report is calculated and appropriate developer is assigned considering features of bug reports and characteristic of old bug that a specific developer has fixed.

iii. Brief Note:

1. *Motivation*:

   Due complexity of maintaining duplicate bug, it has already occupied lot of resources. Once a decision is taken about delicacy of a bug, if it not duplicate, it has to be assigned to developer to resolve it. This is non-trivial process of bug assignments. Automating the process of assigning bugs to developers would save lot of precious time and resources.

2. *Related Work*:

   There is one automated bug triaging processes which has fuzzy set method named Bugzie which recommends fixers for a bug. DREX is one more automated duplicate bug triaging and developer assignment by proposing k-nearest neighbor search methods. Authors have used those tools for basic comparison with their tool DevRec.

3. *Data:*

   For experiments in this paper, author have used 5 large bug report datasets including GCC, OpenOffice, Mozilla, Netbeans, and Eclipse containing a total of 107,875 bug reports. All bug reposrts having status 'closed' or 'fixed' are only used in experiemtns. For statictical analysis of data collected, they have separated data using columns namely 'Reports', 'Resolvers', 'Terms, 'Average Resolvers', 'Product' and 'Componemts'.

4. *Results:*

   DevRec (new tool introduced in this paper) improves the average recall@5 and recall@10 scores of Bugzie (old existing too used by author as base comparison) by 57.55% and 39.39%, respectively. DevRec also outperforms DREX (old existing tool used by author as base comparison) by improving the average recall@5 and recall@10 scores by 165.38% and 89.36%, respectively.

5. *Future Work*:

   While finding out distance between two bugs, authors are planning to explore other text mining techniques so that DevRec can solve Developer Recommendation problem more efficiently. For generalization, authors plan to use more bug reports from more projects.

iv. Ways paper would be improved:

1. Bug resolver is new term coined which state a bug resolver can be anyone who participate/ or contributes in bug resolving. Authors have omitted differentiation between these types of bug resolvers. Participants might not help in getting insight details of bugs for specific developer assignments.

2. Biased assignments were not at all considered while automatic assignment of bugs to developers.

3. In big organizations, at first bugs are assigned to a group i.e. 'webmaster'. Then after filtering little bit, a bug is assigned to a respective developer. Authors did notice such assignments but decided to exclude those bugs from dataset. These bugs should be also considered in order to have more generic solution to Developer Recommendation problem.