Reading Assignment 1

Bhanu Anand, Esha Sharma, Vinay Todalge

i. Reference to the paper:

Chengnian Sun, David Lo, Siau-Cheng Khoo, Jing Jiang. 2011.
Towards more accurate retrieval of duplicate bug reports in Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering.

ii. Important Key-words:

1. Duplicate Bug Report

   A Bug report is summary of bug found in an application. Same bug is reported by different interpretation by different people. Such reports which are refereeing to same bug are called as Duplicate Bug Reports.

2. Triaging Duplicates

   Triaging is process of deciding whether a particular bug is duplicate or not.

3. Duplicate Bug Report Retrieval Function (REP)

   Authors of this papers have come up with function called as REP which tries to retrieve duplicate bug reports efficiently.

4. Extended BM25F

   Extended BM25F is textual based formula used for deciding extend of similarity between two bugs.

iii. Brief Note:

1. *Motivation*:

   While maintaining a huge bug repository for big enterprise application, tracking duplicate bug is potential problem. Similar bug might be selected repeatedly and worked on it unnecessarily. Resolving same bug is unnecessary work leading to wastage of resources. This motivates idea of keeping track of duplicates and filter them to get rid of duplicate bugs.

2. *Related Work*:

   There has been numerous suggestions on how to track duplicates. One of them would be manually filtering duplicate bugs. But considering huge bug repositories, this method is comparatively slow. Automating this process would certainly save time but paradigms used to find duplicate might not be as accurate as where all duplicates are reported.

   One of the first studies for finding duplicate bugs involved natural language processing. Each word in bug report is considered as feature vector while comparing with other bugs. In finding similarity, word frequency of each word is compared against other bug report and accordingly duplicates are decided. Furthermore, in order to increase accuracy in finding duplicates, there has been use of feature like n-grams, inverse document frequency (IDF), probabilistic approach using support vector machine (SVM).

3. *Patterns for Triaging*:

There are certain approached suggested for automating Triaging process.

a. Filter duplicates before reaching triagers: This approach reduces traigers overload and if accurately filtering algorithm is used, it increases efficiency.

b. When a new bug reported, Provide top-k similar bugs for investigation: This approach supports Bettenburg et al's thought that one bug report might only provide partial view of defect, while multiple bug reports complements each other.

Authors of paper had gone with approach where top-k retrieval of similar bug report would accurately report duplication.

4. *Motivation for Retrieving top-k related bugs (REP)*:

   Instead of detecting duplicate bugs, retrieving top-k related bug report would give us better chance of having clear picture of similarities between duplicate bugs. In our opinion, this might lead us to root cause of bug up to certain extend

5. *Motivation for BM25F model*:

   BM25F is model used for checking similarity between two bug reports based on textual representation of bugs. This model is better fit shorter queries. Extended BM25F model supports longer queries for duplicate bug report detection. Apart from this, there are some of characteristics of bug reports which can be used for detecting duplicate bugs such as Component field, Version field, Priority of bug fields etc. Such extra features are used in extended BM25F to accurately detect duplicate bugs.

6. *Data:*

   To implemented new model, authors had used 3 large software bug repositories from Mozilla, Eclipse and Open Office, enough variant data to check wide variety of bugs.

7. *Results:*

   Authors had compared their results with results in paper by A. Sureka and P. Jalote, "Detecting duplicate bug report using character n-gram-based features," in Proceedings of the 2010 Asia Pacific Software Engineering Conference, 2010, pp. 366–374.

   Author claimed that experiments shows 10-27 % relative improvement in recall rate@k and 17-23 % relative improvement in mean average precision over their previous model. More specifically, from 209,058 datasets from Eclipse, they achieved recall rate@k of 37-71 %and mean average precision of 71%.

8. *Future Work*:

   With provision of indexing structure, which speed ups retrieval of duplicates and efficiently maintains huge bug repository, Authors are planning to integrate this model with Bugzilla tracking system.

iv. Ways paper would be improved:

1. Instead of just retrieving top-k similar bug reports, Authors could have put little more intelligence in REP function to find out the root cause of bugs. As after fetching top-k similar bug

reports, we can have enough views/details of same bugs so that finding root cause wouldn't be difficult.

2. Authors have considered pretty much well known Fields in bug report for finding similarity i.e. Summary, Description, and Product etc. In order to make more generic duplicate bug detector, there should be provision of addition of certain fields which are not considered by Authors. Dynamic addition of such fields would save resources for calculation and prevent working on unnecessary blank fields.

3. Main aim was to enhance accuracy in detecting duplicate bugs. Efficiency measures such as Time Complexity, Space Complexity are not mentioned/discussed paper in details. These measures should be discussed in details when there are attempts of working with large real time applications such as Bugzilla tracking system.