Reading Assignment 3

Bhanu Anand, Esha Sharma, Vinay Todalge

i. *Reference to the paper*:
Authors: Xiaoyin Wang, Lu Zhang, Tao Xie, John Anvik and Jiasu Sun
Paper Name: An approach to detecting duplicate bug reports using natural language and execution information in Proceedings of the 30th international conference on Software engineering 2008.

Relationship to first paper: The authors of the first paper quote the current paper in their related work section. The first paper refers to the fact mentioned in this paper that only natural processing similarities between bugs are not sufficient for detecting duplicates.

ii. Important Keywords:

1. Triager: A triager is a person who decides whether a report should be worked on and who should work on it.
2. Duplicate bug report: A duplicate bug report, which has exact same behavior, as that of one of the existing bug report.
3. Execution information: Information of bug-revealing runs that cause a bug report to be submitted is considered as Execution Information. Authors had come up with novel idea to use this information as well while determining duplicate bugs.
4. Information Retrieval: Authors are retrieving relevant information from summary, details of bugs and execution information. This information retrieval involves magical heuristics, which leads to classifying more relevant bugs.

iii. Brief Notes:

1. *Motivational Statements*: Natural language information is containing uncertainty and imprecision. Execution Information contains more relevant and precise information about bugs. Thus execution information helps triager to narrow down search for duplicate with more efficiency and precision.
2. *Related Work*: Authors are concerned about how natural language processing is taken into consideration while detecting bugs. They adopt same technique used by one of initial duplicate detection technique. In addition to that they adopted same natural language comparison technique for finding duplicates, but with addition with execution information.
3. *Data Used:* The authors have used bug repots from bug repositories from Eclipse and Firefox, both having summary and detailed information about bugs. Authors had manually entered execution data for bugs.
4. *New Results*: Authors calibrated approach (Using both Natural language and Execution Information) leads to an increase of 10 to 20% in detecting duplicates and 18 to 26% increase in recall rates. To be precise, with Firefox bug repository this approach can

detect 67 to 93% of duplicate compared to 43 to 72% using old approach using natural language processing.

5. *Heuristics*: Authors found that only one of them (Natural Language processing and Execution Information) cannot be used for finding similarity efficiently among bugs. They had come up with some heuristics which ranks/weights the potential duplicates with combination of both approach. In Basic Heuristics, combined similarity is just average of NLP similarity and Execution information similarity. In Classification Based Heuristics, classes/ranks were assigned to bugs according to value of NLP similarity and Execution information similarity.
6. *Commentary*: Yet author's claim says execution information is equally important as that of natural language information about bugs, their heuristic methods are tending towards in more favor of natural language processing. In Classification based heuristics, highest ranked class is the one, which favors NLP similarity compared to Execution Information similarity.

iv. *Three ways the paper could be improved*:

1. Bug repositories might have number of invalid bug reports. Reason behind invalid bugs might be non reproducible, unable to follow test case, workflow changed etc. Author have straight forwardly discarded those invalid bugs while detecting duplicate bugs. Sometimes due to some minor mistakes while reporting a bug, bug can be considered as invalid bug. A little attention to those invalid bugs should be given in order to have a fully functioning bug detection scheme.
2. The authors have mentioned only 3 techniques for collecting execution information. Bugs that are not covered under none of those techniques will never be analyzed by author's novel idea.
3. This experiment has lack of variety of bug's reports for analyzing bugs. As authors have claimed execution information plays vital role in determining duplicates, each company has different standards/execution environments/application specific workflows. Authors did not mention anything about that. It would be nice to consider those aspects as well in order to have robust bug duplicate detection system.