My research experiments investigate the following research questions:

RQ1: How effective are SBFL and IR fault localization technique individually/standalone?

This question helps my research to understand the performance of widely-used techniques.

RQ2: How effectively can we combine both techniques for better fault localization? Which combination are the best that this research suggest?

This question considers ways of combining different techniques efficiently and effectively, and evaluates the performance of the combined technique.

RQ3: What is the run-time cost of standalone techniques and combined techniques for each program?

The previous question concerns on the fault localization standalone and combined technique’s effectiveness, while this question considers the efficiency of the techniques. The best combination technique is the one that have effectiveness and efficiency balance.

RQ4: How long is the average of execution time for each technique on one bug?

The time cost to localize fault for each bug or program either acceptable or not

RQ5: How effective the combined approach is when compared with the state-of-the-art techniques?

This research question compares the performance of our combined approach

The main contributions of this thesis can be identified as the followings:

* A novel empirical study that compares a specific range of fault localization techniques on real faults which is between SBFL, query and text similarity technique.
* Observation on relationship between SBFL, query and text similarity behavior by using a real-world java Defects4j dataset.
* Proposed a combined technique that are configurable based on time spent, and the accuracy to localize faults performance.
* An infrastructure/architecture/model for evaluating and combining fault localization techniques for future research.

Result and Analysis: Individual/standalone technique.

My research has the following finding (see Table 1 below). Table 1 contains results for overall performance of individually/ stand-alone technique on all 395 faults in Defects4j programs where the boldface indicates the best-performing technique for each program:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Program | SBFL | | | Document Similarity | | | Query | | |
| Accuracy | Time (s) | Time per bug (s) | Accuracy | Time (s) | Time per bug (s) | Accuracy | Time (s) | Time per bug (s) |
| Time | (18/27) 67% | 31. 35 | 1.16 | (14/26)  54% | 74.73 | 2.87 | **(19/26)**  **73%** | 1.50 | **0.06** |
| Mockito | **(25/38) 66%** | 5.26 | 0.14 | (6/37)  16% | 89.42 | 2.42 | (11/38)  29% | 2.49 | **0.07** |
| Lang | (60/65) 92% | 0.73 | **0.01** | (54/63)  86% | 191.66 | 3.04 | **(61/64)**  **94%** | 4.11 | 0.06 |
| Math | (75/106) 70.8% | 4.04 | **0.04** | (59/106)  55.6% | 258.89 | 2.44 | **(84/106)**  **79%** | 14.86 | 0.14 |
| Chart | **(22/26) 85%** | 2.85 | **0.11** | (7/8)  87.5% | 21.75 | 2.72 | **(8/8)**  **100%** | 0.49 | **0.06** |
| Closure | (70/133) 53% | 643.72 | 4.84 | (41/131)  31.3% | 405.37 | 3.10 | **(72/131)**  **55%** | 13.93 | **0.11** |
| Overall performance | (270/395) 68.4% | 687.95 | 1.74 | (181/371)  48.8% | 1041.82 | **2.81** | **(255/373)**  **68.4%** | **37.38** | **0.10** |

Table 1: Overall individual/stand-alone performance of SBFL, query and document similarity technique

* Overall performance for individual/stand-alone performance, both SBFL technique and query technique surprisingly are the same where both score 68.4% of accuracy in fault localization for all six real-world defects4j program. However, time for query technique to complete execute all six real-world defects4j programs are the fastest or the best with only 37.38s time taken compared to 687.95s for SBFL technique.
* Document similarity is the worst performance for both in term of accuracy and time where it scores 48.8% accuracy with 1041.82s time taken to complete all six real-world defects4j programs.
* Even though query technique and document similarity technique are from the same fault localization category, query technique always better than document similarity technique in all six real-world java Defects4j programs with 68.4% accuracy of fault localization compared to 48.8% for document similarity. To complete all six real-world program experiment execution, both query and document similarity technique spend about 37.38s and 1041.82s, respectively.

Figure 1: Comparison on accuracy of overall individual/stand-alone performance

* Query technique also better than SBFL technique in each real-world java Defects4j programs except for Mockito programs where SBFL perform the best for Mockito with 66% fault localization accuracy compared to only 29% for query technique. Query techniques manage to score the Time program with 73% accuracy compared to 67% for SBFL, Lang program scores 94% compared to 92% for SBFL, Math program scores 79% compared to 70.8% only for SBFL, Closure programs score 55% compared to 53% accuracy for SBFL. Chart program score 100% accuracy with only 8 bugs involved from 26 bugs as another 18 bugs did not have bug report reference and could not proceed without bug report.
* SBFL technique are better than document similarity technique for each real-world java Defects4j programs excepts for Chart programs with 87.5% compared to 85% for SBFL technique. However, as mentioned before Chart programs only managed to run 8 bugs as another 18 from the total of 26 bugs does not have any information regarding bug report/link to proceed with. Time program scores 67% for SBFL technique compared to 54% for document similarity, while Mockito score 66% compared to 16%, lang scores 92% compared to 86%, Math program scores 70.8% compared to 55.6% and closure program scores 53% compared to 48.8%, respectively.

Figure 2: Comparison on overall time spent for each bug in individual/stand-alone technique

Result and Analysis: Combine/Hybrid technique

Another finding that are analyzed from the combine/hybrid experiment are presented in Table 2 below. Table 2 shows the result performance of combination/ hybrid techniques on all 395 faults in all six real-world program Defects4j programs where the boldface indicates the best-performing hybrid technique for each program from accuracy perspective. If the accuracy is similar, then the time cost will be included as weight in deciding which technique are the best:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Program | SBFL + Document similarity | | | SBFL + Query | | | SBFL + Query + Document similarity | | | Query + Document similarity | | |
| Accuracy | Time (s) | Time per bug (s) | Accuracy | Time (s) | Time per bug (s) | Accuracy | Time (s) | Time per bug (s) | Accuracy | Time (s) | Time per bug (s) |
| Time | (21/27)  78% | 106.08 | 3.93 | (24/27)  89% | 32.85 | 1.22 | **(25/27)**  **93%** | 107.58 | 3.98 | (23/26)  88.46% | 76.23 | 2.93 |
| Mockito | (25/38)  66% | 94.68 | 2.49 | **(28/38)**  **74%** | 7.75 | 0.20 | (28/38)  74% | 97.17 | 2.56 | (12/38)  31.6% | 91.91 | 2.42 |
| Lang | (62/65)  95% | 192.39 | 2.96 | (64/65)  98.5% | 4.84 | 0.07 | **(65/65)**  **100%** | 196.5 | 3.02 | (63/64)  98.4% | 195.77 | 3.06 |
| Math | (82/106)  77.4% | 262.93 | 2.48 | (98/106)  92.5% | 18.9 | 0.18 | **(100/106)**  **94.3%** | 277.79 | 2.62 | (91/106)  85.8% | 273.75 | 2.58 |
| Chart | (23/26)  88.5% | 24.6 | 0.95 | (**23/26)**  **88.5%** | 3.34 | 0.13 | (23/26)  88.5% | 25.09 | 0.97 | **(8/8)**  **100%** | 22.24 | 2.72 |
| Closure | (87/133)  65.4% | 1049.09 | 7.89 | (97/133)  73% | 657.65 | 4.94 | **(104/133)**  **78.2%** | 1063.02 | 7.99 | (84/131)  64% | 419.3 | 3.20 |
| Overall performance | (300/395)  75.95% | 1729.77 | 20.7 | (334/395)  84.56% | 725.33 | 6.74 | **(345/395)**  **87.34%** | 1767.15 | 20.27 | (281/373)  75.34% | 1079.2 | 16.91 |

Table 2: Overall performance of combination/ hybrid techniques to all programs

Figure 3: Comparison on accuracy of overall combination technique performance

* For combination/hybrid technique, the best performance on accuracy is the SBFL technique that combines with query and document similarity techniques with 87.34% accuracy on fault localization where 345 bugs are managed to be allocate from 395 bugs. Though, the total time to perform this combination technique on all 395 bugs is a lot which is 1767.15 seconds however, when average calculation for each bug executed are being made, the total average time taken on each bug are around 20.27 seconds per bug which is count as acceptable.
* However, if time is one of the concerns without jeopardizing accuracy, combination of SBFL and Query technique is the best in term of fault localization accuracy where it scores 84.56% with 334 bugs are managed to be allocate from the total 395 bugs. Plus, the time taken for overall programs is 725.33 seconds, the fastest among all combination technique where it only took 6.74 seconds per bugs to be executed.

Figure 4: Comparison on overall time spent for each bug in combination technique

* The combination of document similarity with SBFL technique results does not have significant difference with the combination of document similarity with query technique where both scores 75.95% and 75.34% respectively.
* On the other hand, the time taken to execute all six programs for combination of document similarity and SBFL is 1729.77 seconds with average 20.7 seconds time taken for each bug to be localize while the combination of query and document similarity technique spend 1079.2 seconds to execute all programs with average 16.91 seconds per bugs to be allocate.
* As shown in the table, most programs perform the most with the combination of three technique which are SBFL, query and document similarity where Time program managed to score 93% while Lang program scores 100%, Math programs scores 94.3%, and Closure program scores 78.2% of accuracy to localize fault.
* While for Mockito program, the best performance is the same for both combination of SBFL with Query and combination of SBFL with Query and Document similarity where both scores 74% in accuracy to localize fault. Now this is the time when the time taken to execute the program will be helpful. Since it only took Mockito program about 7.75 seconds with average time 0.2 seconds per bugs to localize faults by using SBFL and Query technique, this combination performance is the best compared to the combination of SBFL with Query and Document similarity where the time spend is 97.17 seconds with average 2.56 seconds per bug to localize faults.
* For Chart program, the accuracy result for the combination of document similarity and SBFL are the same with both the combination of SBFL and query technique, and the combination of SBFL, query and document similarity where all three technique scores 88.5%. From time perspectives, the combination of SBFL and query technique is the fastest where it only took 3.34 seconds to run the whole Chart program in fault localization with average 0.13 seconds per bugs.
* The combination of SBFL and document similarity technique took 24.6 seconds to localize fault with average 0.95 seconds per bugs while the combination of SBFL, Query and Document similarity took 25.09 seconds to localize faults with average 0.97 seconds per bug.
* Though the combination of query and document similarity technique managed to get 100% performance in accuracy, but as mentioned before this, there are missing bug report for 18 bugs in that programs so only 8 bugs are included.
* If we are looking at the time cost perspective, the combination of SBFL and Query performance are the best with Time program only took 32.85 seconds with average of 1.22 seconds for each bug to localize fault, Lang program took 4.84 seconds with 0.07 seconds for each bug to localize fault, and Math program took 18.9 seconds with 0.18 seconds for each bug to localize faults.
* While for closure program, the best combination technique in time perspective is the combination of Query and Document similarity where it took 419.3 seconds with 3.20 for each bug to localize faults compared to other combination technique.

As a conclusion from the experiment results, though usually time spent for standalone/ individual technique are slightly shorter than the combined one, the combination or hybrid technique effectiveness in fault localization significantly outperforms standalone techniques. In my opinion taking time less than 10 seconds for each fault executed are acceptable as long as the performance of the fault localization accuracy are at the best. I also want to highlight the important of quality bug report where there are cases that fault location unable to be identified as the information provided are not enough. I hope that this research might be complementary to existing techniques as it could further improve state-of-the-art technique by combining with existing techniques.