

Thanks to all the reviewers for useful comments and suggestions.

Reviewer\_1-Q1, We did not include more recent bug localization techniques, because they are not closely related to our work. For example, 1) Amalgam is based on five components – four of them except version history are not considered in our proposed approach (i.e., BLuAMIR), 2) BRTracer consider segmentation and stack trace analysis, which we don't, 3) BLIZZARD is a query reformulation based bug localization technique, while BLuAMIR does not focus on any kind of query reformulation.

Reviewer\_2-Q2, We normalize the association score by dividing the association score of a given source file and the maximum association score calculated from all source files for a given query bug report.

Reviewer\_2-Q3: For a given training dataset, we need to create the association map once. So, even if takes a while to analyze the bug-fix commit history, it does not require to be repeated.

Reviewer\_3 and Reviewer\_2-Major: We are well aware of the problem with 10-fold cross validation, we performed this validation technique on our dataset to test whether our proposed approach is working or not. However, in practical there is no such way of learning from future instance. So, there will be no risk in this regard.

Reviewer\_1-Q2: In BLuAMIR, we combine our implicit association score with lexical similarity score (i.e., VSM technique). As BLuAMIR already has a component of one (i.e., VSM) of our baseline techniques (i.e., VSM, LSI), we separated the baselines and state of the art techniques, to ensure a fair comparison to check the applicability of BLuAMIR.

Reviewer\_1-Major: We get the bug report (i.e., title and description of a bug) from several bug tracking repository such as <https://bugs.eclipse.org/bugs/> and then we go through all commit messages and collect the changeset for each bug report.

Reviewer\_2-Q1 and Minor: In the motivating example, the  $S_{LS}$  score is not different for VSM and BLuAMIR. Note, that both techniques retrieve same source file, *JaveCore.java* having same score of 0.74 (2<sup>nd</sup> rank in VSM, 4<sup>th</sup> rank in BLuAMIR). Here, BLuAMIR successfully retrieve a relevant source code file (i.e., *CompletionEngine.java*) in 1<sup>st</sup> rank, which is not present in Top-5 results retrieved by VSM. However, total score is calculated as  $S_{Total} = (1 - \alpha) * S_{LS} + \alpha * S_{Assoc}$ . For example, for the file *CompletionEngine.java*  $S_{Total} = (1-0.4)*0.67+0.4*1.00 = 0.80$ , while considering  $\alpha = 0.4$ .

Reviewer\_2-Minor: We choose the same four open source projects, which are also evaluated by three state of the art techniques. We include all bug reports from two of datasets (i.e., SWT and ZXing). We could not collect all source code files for other two subject systems (i.e., Eclipse and AspectJ). Therefore, we discarded all those bug reports, whose associated change source files are not present in the codebase.

Reviewer\_2-Minor: Comparison between Baseline VSM and BLuAMIR (for AspectJ, SWT and ZXing) as follows. Note that this is a part of Table 5 where data is shown for Eclipse systems.

System	#Bugs	Approach	Hit@1	Hit@5	Hit@10	MRR	MAP
AspectJ	244	VSM	29.10%	52.45%	61.06%	0.38	0.22
		BLuAMIR	<b>33.20%</b>	<b>54.92%</b>	<b>66.39%</b>	<b>0.43</b>	<b>0.23</b>
SWT	98	VSM	42.71%	73.96%	86.46%	0.57	0.49
		BLuAMIR	<b>45.93%</b>	<b>75.00%</b>	82.29%	<b>0.58</b>	<b>0.50</b>

ZXing	20	VSM	55.00%	75.00%	80.00%	0.65	0.56
		BLuAMIR	55.00%	<b>80.00%</b>	<b>85.00%</b>	<b>0.67</b>	<b>0.62</b>

According this table, most of the cases (13 out of 15) shown in bold, BLuAMIR outperforms baseline VSM, in one case it is preserved and only one case it is worsen. However, for Eclipse (i.e., Table 5) BLuAMIR improves for all 5 cases. So, these results provide a clear picture of better performance by BLuAMIR than baseline VSM.

Reviewer\_2-Major: In page 6 (Table 5), the improvements is calculated as

Increase= $((\text{New\_number} - \text{Original\_number}) / \text{Original\_number}) * 100$ .

So, for Hit@ it is =  $((27.45 - 23.09) / 23.09) * 100 = 19\%$ . Thus, BLuAMIR shows 19% higher (i.e., improvement) in Hit@1 than baseline VSM.

Reviewer\_2-Major: Regarding Table 8, we missed to compare the result rank improvement with BLuIR, but will do it in the paper if it is accepted. However, this are the results for BLuIR:

Reviewer\_2-Major: We missed to mention these exceptions that in Figure 7, Zxing shows decrement in MRR performance but in Figure 8 shows increment in MAP performance. On the other hand, in Figure 8 AspectJ shows decrement in MAP performance but in Figure 8 shows increment in MAR performance.