

1. Eclipse bug reports finding website: <https://bugs.eclipse.org/bugs/>
2. Firefox bug reports finding website: <https://bugzilla.mozilla.org/home>
3. Mobile bug reports finding website: <https://bugzilla.mozilla.org/home>.

All the pairs are sorted according to their similarity score. We have taken to take 50 bug reports (for which enough information is available) from each dataset. Three datasets in total (Eclipse, Firefox, Mobile) and two types (Textually similar and Textually dissimilar). All the bug reports are collected in google sheets.

Bug ID and Duplicate ID are given in these google sheets named manual analysis for similar and manual analysis for dissimilar. You can search the bug id and dup id using the URLs mentioned above by bud id and dup id and find all the necessary information regarding the bug reports. You must compare the bug id with its corresponding duplicate id based on three things in the description. 1. Observed behavior 2. Expected behavior, and 3. Step to reproduce (and lastly, overall). Sometimes these are explicit in bug reports; sometimes, we can find all in the description or comment together.

If we find any similarity (textually), we will put Similar = 1 in the excel cell and Not similar = 0. Then based on all three aspects (or if not all of these are available, then just the overall conclusion), the overall will be denoted as 1 = Similar or 0 = Not similar.

If any information is not found, then just leave it blank. We will collect more bug reports in this case.

If you notice any specific pattern or repetitive behavior in one type of dataset (for example, textually similar), please mention it in the excel sheet and give overall feedback on textually similar and dissimilar bug reports here. Also, note down the time to complete.

Check for related words, components, versions, and technology, and use your best judgment.

Example

Textually Dissimilar Duplicate Bug Reports (Dataset: Mobile system)			
Bug ID: 1618582		Bug ID: 1618579	
Title	Description	Title	Description
Dynamic toolbar should not hide when the page is not scrollable.	<p>EB: I think this just boils down to Fenix should not hide the toolbar when a page is not scrollable.</p> <p>S2R: Load the test case in Fenix Nightly and scroll down so that the dynamic toolbar disappears; observe that the grey overlay doesn't cover the bottom part of the Screen.</p> <p>OB: Elements with height: 100% don't include the dynamic toolbar.</p>	Text selection caret is misplaced with the dynamic toolbar.	<p>EB: Text selection caret needs to update its position.</p> <p>S2R: Load a minimal example in Fenix Nightly, ensure the URL bar is not visible, select text, and observe text selection carets are shifted up from where they usually should be. Long tap to select any word in the test case. Swipe up to make the dynamic URL bar disappear.</p> <p>OB: The text in the test case is in a 'position: fixed; bottom: 20px' element, so this looks similar to bug 1611032. No scroll or app-related callbacks are observed, so AccessibleCaret doesn't seem aware of something being changed.</p>
Keyword Overlaps: 3			