Tools for removing JS dead code

ID	URL	Is the tool available?	Is the tool usable in the experiment?
sQiong	An Optimization Method of Javascript Redundant Code Elimination based On Hybrid Analysis Technique	No	No, since there is no tool we can run.
Vazquez	Slimming javascript applications: An approach for removing unused functions from javascript libraries (Vazquez et al.)	Yes https://github.com/hcvazquez/UFFRemover	Yes. The tool requires six steps: 1) instrument the js file for which we wish to identify and remove unused foreign functions (a command is given within readme) 2) replace the original file with the instrumented file within index.html 3) run the website using http-server and use it (click on all possible actions) 5) save the content from the browser console and run the optimise command (the command is given within readme) 6) the output gives a new optimised file and the metrics about the optimization are printed in the console IMPORTANT, the above process can be automated.
JSLIM	JSLIM	No	No. However, while searching for the tool described in the paper we identified this tool (https://github.com/zgrossbart/jslim, which (despite having the same name) is not the same tool mentioned within the JSLIM paper
Muzeel	Muzeel: Assessing the Impact of JavaScript Dead Code Elimination on Mobile Web Performance	Yes https://github.com/comnetsAD/Muzeel	Yes. The tool requires the following steps: 1) Set up a MySQL database with a certain structure 2) In each proxy .py file modify the load function to use the credentials for the MySQL database
Goel	Utkarsh Goel and Moritz Steiner. 2020. System to Identify and Elide Superfluous JavaScript Code for Faster Webpage Loads. arXiv preprint arXiv:2003.07396 (2020)	No	No since there is no tool we can run.
Lighthouse	Google Lighthouse - Remove unused JS	Yes, in the Chrome WebDev Tools	No. We decided to do not use Google Lighthouse in this evaluation. The main motivation is that the level of granularity of Google Lighthouse is at the Javascript file level (instead of individual JavaScript functions); this would result in (i) comparing tools with different purposes and completely different level of abstraction and (ii) an unfair comparison with respect to the all the other tools (including Lacuna), which are more fine-grained and are able to produce more accurate results.