1. How you should create the benchmark/input for XFinder tool?

Benchmark is a set of reviews, each review includes a set of submitted files for review process .

XFinder tries to recommend a list of reviewers for each submitted file related to each review.

To create the benchmark we first have to choose the appropriate reviews. For example I selected those reviews which at least 50% of files submitted for that review are not test files.

2. Steps for creating input for XFinder:

* Create a text file for each review id in the benchmark and set the name of the text file as review id.
* Select the name and path of each file from the list of submitted files related to each review id. These should be written in separate row in the text file.
* All the text files should be in one folder.

3. Brief explanation for important Java files inside the package:

* DevRecommender.java has the void main function. Input and output path should be set in this file.
* Dictionary.java: SAXparser for code review log
* DictionaryGitLog: SAXparser for commit log
* XFinder: Does the main work of this program. Finds the reviewers who reviewed a file that contained the desired file/path, then it finds the xfactors for each reviewer

4. How you should run the tool? Run DevRecommender.java

* DevRecommender.java. There are three variables in this file which should be set.
  + @INDir is the input folder (as I explained before)
  + @OUTDir is the output folder
  + @ReviewLogDir is the directory of code review log. It doesn't matter if you run the tool for code review history or commit history. For both cases this variable should be set as code review log path.
* XFinder.java Calculates the xfactor for each file inside the input text file.
  + @interactionDictXMLDir is the directory of the code review log/Commit log. When running the XFinder based on code review, @interactionDictXMLDir should be set as the directory of code review log and when running XFinder based on commit history, @interactionDictXMLDir should be set as the directory of the commit log.
  + As I said above, Dictionary.java and DictionaryGitLog.java are two SAXparsers for parsing the review log and the commit log. If we want to run the XFinder for code review we should have this line in XFinder.java:
    - Dictionary Dict = new Dictionary(interactionDictXMLDir);
  + and if we want to run the XFinder for the Commit log the above line should be commented and this line should be uncommented:
    - DictionaryGitLog Dict = new DictionaryGitLog(interactionDictXMLDir);

5. Not all the HashMaps from Dictionary.java and DictionaryGitLog.java have been used in this project. Some of them are used in different projects.