Option B (Empirical evaluation)

What is the problem? Why is it an important problem?

Every change requests a deep understanding of the code and an ability to relate the concept of the change to software artifacts, which are source codes. This task can be challenging for new developers in a team and experienced developers in large software. In this study, we propose a deep neural network that, based on discussions in a GitHub issue about a change to the software or bug fix, will identify the potential files that need to be changed.

Brief review of previous work concerning this problem

Some studies [1,2] use information retrieval methods for concept location to find the best match in source files based on a user query. The downsides of this method are:

- The user should type queries instead of natural language descriptions.
- It heavily depends on the words chosen by developers in the source code instead of the real meaning of the code.
- The precision of these models is very low, less than 40%.

As far as we know, no one uses neural networks to solve this problem.

What is the intuition behind the technique that you have developed?

A software developer with a good understanding of the code can guess which files should be changed to add the feature or fix the bug discussed in a reported issue. Thus, having a model that understands the natural language and code language may be able to relate these two artifacts.

Describe the technique that you developed

The first step is to prepare the dataset, which will have three columns. The first column is a discussion in a GitHub issue, the second column is the content of a source file, and the third column will be a flag indicating that the file is related to that discussion. All of these data could be retrieved using the GitHub API.

A bidirectional LSTM model will be used to learn the natural language in the discussion (first column) and the source file (second column). We will build our network on top of Bert [3] and code2vec [4]. Finally, a fully connected neural network will do the classification.

[1] Marcus, Andrian, et al. "An information retrieval approach to concept location in source code." *11th working conference on reverse engineering*. IEEE, 2004.

[2] Poshyvanyk, Denys, and Andrian Marcus. "Combining formal concept analysis with information retrieval for concept location in source code." *15th IEEE International Conference on Program Comprehension (ICPC'07)*. IEEE, 2007.

[3] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

[4] Alon, Uri, et al. "code2vec: Learning distributed representations of code." *Proceedings of the ACM on Programming Languages* 3.POPL (2019): 1-29.