4 Design

4.1 Design Context

4.1.1 Broader Context

Area	Description	Our Answer
Public health, safety, and welfare	How does your project affect the general well-being of various stakeholder groups? These groups may be direct users or may be indirectly affected (e.g., solution is implemented in their communities)	Our tool will lead to more quality datasets for developers working on code-to-text applications. This goes on to their applications, possibly making society better as a whole.
Global, cultural, and social	How well does your project reflect the values, practices, and aims of the cultural groups it affects? Groups may include but are not limited to specific communities, nations, professions, workplaces, and ethnic cultures.	Our work is tied to a more specific community, that being the development community. Worldwide effect will be made by machine learning eventually, and our tool will be a very small subset in that.
Environmental	What environmental impact might your project have? This can include indirect effects, such as deforestation or unsustainable practices related to materials manufacture or procurement.	Increased energy usage by a deep neural network.
Economic	What economic impact might your project have? This can include the financial viability of your product within your team or company, cost to consumers, or broader economic effects on communities, markets, nations, and other groups.	Our project may save machine learning engineers time (and therefore money for their org) by providing better datasets and a way to obtain datasets.

4.1.2 Prior Work/Solutions

– There are existing projects that make up components of what we are trying to accomplish. There is an existing project called 'jflex' that we have utilized to get an idea of how we are going to break up and label our source code. We have also used some preexisting datasets of chunked code as a baseline for what our datasets could look like.

– In the aforementioned project 'jflex' we have used that to our advantage to see one method of what we want to accomplish. The disadvantage is that the existing project is not as granular as we want it to be, and we will have to use our own method to match our specs.

Pros

- Get an idea of how we will use regular expressions to assign labels to fragments of code
- We can compile a nice dataset to compare against the dataset we will eventually use

Cons

- The existing project does not use our exact labels and it is a general idea of the final product
- We will have to create our own parser using different regular expressions to get more granular labels.

4.1.3 Technical Complexity

Our project is technically complex as it pertains to analyzing code that is written by humans, chunking it in possibly numerous differing ways. This chunked code will be used as data for a machine-learning algorithm. Analyzing code is quite a complex task due to the numerous ways that the code could be chunked and the vast amount of edge cases that java has. Machine learning algorithms are not trivial either since there are or always will be numerous ways to accomplish the same goal. Different programming patterns can be used to achieve the same or similar results. Once both tasks are completed, we will need to examine the results and reexamine the chunking method to maximize results.

Both things combined, coupled further with the reiteration to ensure a quality product, are going to make this technically complex.

4.2 Design Exploration

4.2.1 Design Decisions

Language

Due to this project being centered around programming, we needed to determine what language we should use since it would be unproductive, to say the least, if we all used different ones. Our client narrowed down the selection significantly by saying we should use either java or python.

Chunking

This project's final output includes chunking existing code into a more condensed syntax, which will later be used by the client. As a result, this decision cannot be made lightly and will most likely include many iterations and trying multiple options and may even include some capability to alternate between them depending on further direction from client.

4.2.2 Ideation

Chunking

Since this particular design decision is vital to the project's success and outcome, the potential options were brainstormed with us and the client. We identified six different ways to chunk the code.

Word by word

Creating a chunk for every single line of code.

Common Phrases

The most abstract one on the list and probably the most complex, but it would identify common programming patterns and phrases such as a guard clause or something on a broader scale, such class inheritance.

Locality

This would chunk the code based on the variable locality. For example, there is the file locality, class, function, if and for loop localities to be considered.

Functions

Like locality but for just the functions in the code.

One line

Chunking each line and it function such as for loop start or assignment or decrement etc.

Many lines

Like one line but would attempt to group similar lines since often times lines next to each other are often similar in purpose.

Language Decision

When deciding between using java or python source code to create labels, the biggest thing we compared is the syntax. When it comes to creating an algorithm to label source code the more well defined and structured the programming language's syntax is, the easier it appears to be to define an algorithm to parse through source code and create labels. Due to the structure and more symmetrical syntax that java requires, we determined java source code would be easier to label with an algorithm.

4.2.3 Decision-Making and Trade-Off

Thus far we have not had large or complex decisions to make as our project has been very dynamic thus far. Which form of text chunking to use will be a future decision that is decided upon the results of our machine learning model results. Due to this we have decided to use multiple text chunking methods that are defined in the previous section that we can test and decide upon later.

The main decision that we have made is choosing to use java or python source code. The decision process we used to do this was by getting together as a group and discussing/debating the programmability of chunking java or python source code. Through this extended group discussion, it was determined that due to the structure and symmetry of the Java programming language it would be easier to chunk out. Things like line statements ending with a ';', if statement conditionals being contained in '()', and loops being contained in '{}' are all examples of syntaxial advantages that the Java programming language has that are not required in python. Through our group discussion/debate over the two options it became clear that Java's syntax would be very advantageous to the chunking of source code and made the decision easy to go with Java.

REFERENCES

Klein, Gerwin. *JFlex*, https://www.jflex.de/.

"Hugging Face – the AI Community Building the Future." *Hugging Face – The AI Community Building the Future.*, https://huggingface.co/datasets?search=code_x_glue.