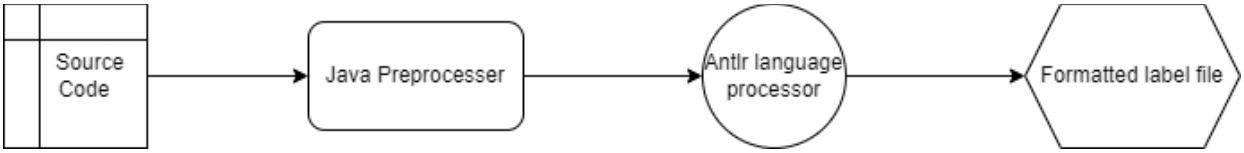
## 4.3 Proposed Design

### 4.3.1 Overview

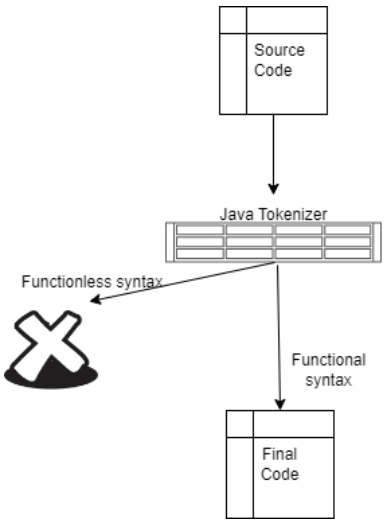
We are creating an input program that allows users to give the program a file, that will then send the formatted output to a text file(tentatively). We take the user input and send it through Antlr which is an open-source language processor. Antlr will parse the file and add Natural Language tags. We will format the output in a machine learning friendly way and send that on to our dataset storage.



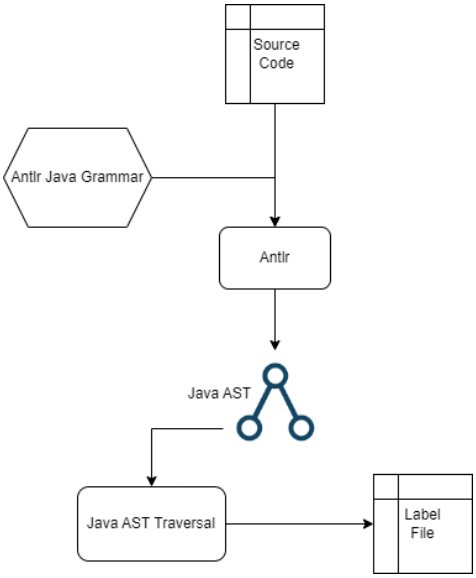

### 4.3.2 Detailed Design and Visual(s)

#### 4.3.2.1: Java Preprocessor

This sub-component is currently under consideration and may make it into the final project. For some Java code, we may have to preprocess the file to make sure anything not serving a core functionality to the language processor will not make it to the language processor. An example would be whitespace lines or comments. Our current idea is using a tokenizer to obtain each token and re-building a Java file with a space as a delimiter. While more efficient ideas are being worked on, this is the simplest solution possible right now.




#### 4.3.2.2: Antlr Language Processor




Antlr is an open-source language parser and lexer that will be critical in obtaining "human-recognizable" parts of code, such as loops and conditionals. We must obtain the AST (abstract syntax tree) from Antlr and parse its output in a data structure. From there, we can build a new output based on the human recognizable features Antlr picks up.

#### 4.3.2.3: Formatted Label File

The output of the Antlr processor will be put into a label file. One decision we're deliberating on is the data output. We must first analyze what data type will be easiest to put into a machine learning program.

### 4.3.3 Functionality

Upon the completion of our system, a user will be able to input the data that they wish to be chunked. The data in question is intended to be files of code, they would then specify the level at which they would want

it to be parsed. The user would specify the level at that they would want their data to be chunked which there are multiple levels; word by word, line by line, common phrases, locality, functions, or single line. The system upon receiving the data and the method in which to parse it will then turn the data into a dataset that is intended for use of a machine learning algorithm.

### 4.3.4 Areas of Concern and Development

Throughout the semester, a lot of time has been spent directly with our users to hash out the fine details of our project design. Currently, the design plans we have set in place will satisfy and meet all of our user's basic and highest-level needs. One of our biggest concerns for delivering this product is going to be the Machine Learning pipeline implementation. Our group has had little experience with the machine learning process, so we are expecting there to be a little bit of a learning curve when it comes to the implementation. We have discussed this concern with our stakeholders and are doing R&D in the meantime to prepare the best we can.

## 4.4 Technology Considerations

This project is being written in Java. Every member has experience using java and it has plenty of support and tools available online. One of those tools was initially Jflex which is a scanner generator. It's written in java for java and was used for generating a scanner used to analyze the code we were looking at. However, it proved time consuming to write the multitude of regular expressions necessary for what we needed. Instead, we moved on to Antlr, which is a parser generator which covers the scanning function we were previously using along with some more features that provide extra flexibility. It is a complex tool that does have a decent learning curve, but the grammars for the code we want to analyze have already been made, so that cuts down on a lot of time investment there.

## 4.5 Design Analysis

Recent setbacks in technology have set us a bit behind schedule in our technology considerations and planning, but after a critical look at the technology we were previously working and researching better solutions, we've found a more solid footing and a clearer idea about what our next steps are. We are more confident that our design will work. We will soon get more answers about the long-term success of this technology and what our data and machine learning pipeline will look like. We've started on a primitive version of the Antlr language processor.