

Journal Report 11

11/25/19-12/9/19

Bryan Lu

Computer Systems Research Lab

Period 2, White

Daily Log

Monday, November 25

Fixed Github issues, prepared README file, and pushed all previous work onto Github in preparation for break.

Monday, December 2

Restructured `relations.txt` and fixed test cases, read original pdf to figure out how to

Tuesday, December 3

Found original project Github, downloaded the files from the original project, and started looking through their code to see how to use it to my advantage.

Thursday, December 5

Continued reading through original project code, and figured out what a decent number of the python classes/methods used do.

Timeline

Date	Goal	Met
11/18	Build the model with <code>scikit</code> , tweaking previous steps as needed, and finish the necessary test cases.	I didn't have any time this week to start building the actual model, but my test cases are now more or less where I want them to be.
11/25	Build, test, and train the logistic model, given the test data and the features computed.	No, Thanksgiving break – did not do work on the project over break.
12/2	Refine logistic learning model with extra methods to try to increase accuracy, and begin extracting the most likely literals.	Unclear how to do this from my work so far, found a different approach to continue work.
12/9	Inject my problem statements, lexicon, and files into the existing framework given by the original project.	N/A
12/16	Harness the existing code to identify the valid logical literals in a problem from the statement alone.	N/A
Winter Goal	Be able to output a set of possible literals (statements) based on detected relations in the problem.	N/A

Reflection

My plan is drastically changing again, because of a major development in my project's progress. I found the Github associated with the paper that my original paper is based off of, and I'm now shifting gears to looking through their files and modifying it to my needs. So far, I've found their project's lexicon, and I've found that they've created actual classes and objects for geometric objects, which is something I've avoided doing because it was a lot of work. Most of the work is now done for me, so all I have to do is add in any object that is not already in my own `relations.txt` file.

I don't currently know how to access and modify the problems they used for training, as they are stored on some sort of server somewhere, but that is something I will have to figure out this week. They specify something called "annotations" for every problem which I assume is similar to what I've been doing for the past two-three weeks for my own problems, that is, marking them up with the appropriate relations per problem. I do know that their project uses JSON requests to store their problems, and I need to format my problems in this way in order for their code to work.