Journal Report 18

03/02/20-03/09/20
Bryan Lu
Computer Systems Research Lab
Period 2, White

Daily Log

Monday, March 2

Worked on adding test cases, formatted annotations and cases in conjunction with creating shell code for running the model.

Tuesday, March 3

Super Tuesday. Voted in the Virginia Democratic Primary.

Thursday, March 5

Added another test case, worked on getting training code to run with the formatted annotations and syntax parses.

Timeline

Date	Goal	Met
2/17	Finalize annotation data for the olympiad problems and ensure that they produce valid, connected semantic trees.	Not sure if this latter part is a concern, but I've decided on a test set I want to get the rest of the process to work on and began annotating.
2/24	Annotate about half of the olympiad problems used as test cases in the correct format.	Difficult to get started, but I've got enough to work with for the next week.
3/2	Start using the Naive Tag Model with olympiad problems, fix any issues that may arise, add more cases.	Making good progress, still working on getting the original project code to take the training data in the right format.
3/9	Train the model, look into any deeper issues, continue adding more cases as needed.	N/A
3/16	Continue training the model, add more cases, start getting some results back.	N/A

Final Goal

Grade	Requirements
A	With a well-trained Tag Model (at least 15-25 problems of experience), allow a user to enter an olympiad geometry problem and get Asymptote code in return. Well-written paper, clear presentation, organized Github with important elements commented for others to pick up work.
B	The above, but with issues, such as: conversion from literals to Asymptote code not working, model not trained well enough, hard-coding for a specific problem. Paper, presentation, or Github lacking in at least one-two aspects.
C	Attempts to train a Tag Model, unable to produce Asymptote code. Poorly-scrapped together paper and presentation, Github disorganized and unreadable.

Reflection

Not much to talk about or report on this week – I'm still working on making the code that allows the model to run test cases fully functional. This is something that I'm reverse-engineering, based on how the code reads in the annotations and recognizing how different sentences within a problem are treated in the analysis.

One problem that I anticipate in the coming weeks is that I haven't got enough annotations for a problem – in particular, in the case where the code can't identify what an object is based on context clues. This might make it so that the semantic tree that is created from the annotations is disconnected and any further analysis will give errors for a particular sentence, or that the tree is too simple and the proper analysis can't be performed with nontrivial results. I'm not sure if this will actually happen, but I certainly might anticipate it.