

## Journal Report 13

01/06/20-01/13/20

Bryan Lu

Computer Systems Research Lab

Period 2, White

---

### Daily Log

#### Monday, January 6

Dusted off my code, tried tracing through `annotations_to_semantic_tree.py` to reverse-engineer the form of the annotations.

#### Tuesday, January 7

Tried tracing through code, consulted Dr. White on recursive descent parsing, made plans on how to move forward.

#### Thursday, January 9

Created my own short script to run annotations parsing on a particular question, started process of fixing errors, modified Python 2 → Python 3 incompatibilities, and debugging. Emailed original article authors for annotations.

### Timeline

Date	Goal	Met
1/6	Figure out what the best next steps are for moving forward, based on not quite reaching my winter goal.	I think I have a fairly concrete plan for what to work on moving forward, but I'm not sure if it's super feasible... need to discuss still.
1/13	Get a script heavily based on their code to be runnable, pending annotations.	N/A
1/20	Adapt the question input for their code to access local files instead of a database, try running their code with olympiad problems.	N/A

## Reflection

This was the first week after break, and coming back after break, I think I needed to fully flesh out what I needed to do with the trove of code that I pulled from the original project github that I recently discovered.

Generally, this is the plan I've come up with to use this code as effectively as possible.

1. *Get syntax parsing to work on a problem* – (this week)
2. *Get a semantic tree from a problem*, assuming I can get my hands on those annotations for the problems or can figure out how they're formatted (this week/next week)
3. Finish modifying the ontology (something I started before break, not super essential and can be done over time)
4. *Fix input from database to have code pull from my local files* (next week)
5. *Start trying to run the Naive Tag Model on my olympiad geometry problems* (training the model now)
6. *Test the Naive Model on a whole bunch of problems and see how it does.* Includes decisions on whether or not I can improve my code by fixing/running their more sophisticated processing model on olympiad problems, or just increasing my data set (which is pretty labor-intensive).
7. *Get predicted literals to output for a given problem.*
8. *Interpret literals and give a set of Asymptote commands.* For whatever literals I get, I figure out what points are in the diagram, and essentially loop through the set of literals and interpret them by drawing what they say. Hopefully this set can be ordered in a meaningful way to make things easier.
9. profit

I'm not sure how plausible this is for the amount of time I have remaining, but I think this is a pretty reasonable path forward.