

Syslab Research Journal 0

BRYAN LU

August 27, 2019

1 Three-Sentence Summary of Project

Geometry problems usually require drawing a diagram in order to begin solving them – however, for complicated geometry problems, drawing an accurate diagram requires quite a bit of artistic ability or a ruler and compass to which not everyone has access. Computers can draw very accurate diagrams using Asymptote code, and can also be taught how to parse geometric problem statements using natural-language processing algorithms. My project will use supervised learning to perform natural-language processing on geometry problems and output a diagram corresponding to the problem statement.

2 Summer Progress

I did a little bit of work over the summer to attempt to scrape olympiad geometry problems off of the Art of Problem Solving website, where I know a lot of these problems are cataloged. This was a little difficult, as the problems are stored as forum posts that are loaded onto the page a few seconds after the page is loaded, so I couldn't automate my code to scrape the problems off of the pages fully. I'm currently debugging the string parsing function that is being used to clean up the LaTeX code embedded in the problems, which will prepare them for use in training my code.

3 Possible Obstacles

My biggest obstacle will be to create a classifier that needs to, at some point in the process, identify what relations there are between the identified geometric objects in a problem statement, based on what relations are detected in the problem statement. This could present a challenge because the way I plan on doing so is by allowing the computer to train itself on test cases, which may become problematic if it returns faulty results.

In general, I will likely face problems having to do with accuracy after training my neural network, because I will have a very small number of test cases to use as I don't have a readily available immense source of geometry problems at my disposal, and I'm not exactly sure where to find one.

4 First Marker of Success

My first marker of success will be when I've produced code to identify whether or not a possible relation between objects is correct, given several features of how the geometric objects appear in the problem statement.

This should be done probabilistically, based on what data the computer has seen before. If I'm able to do this, I can progress to thinking about constructing the logical statements that might appear in the problem based on how the objects are related to one another, and refining what kinds of relations between objects might appear in the text that aren't as obvious as compared to ones I've identified. Identifying relations between should be a strong indication of success, because it allows me to build more structure on top of that algorithm that will get my code closer to "processing" the problem statement correctly.

5 Materials

I don't think I will need any extra physical materials to carry out my project.