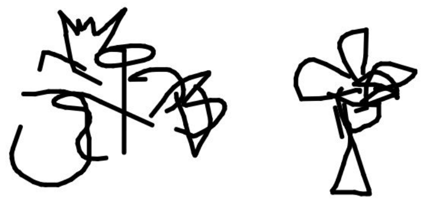
Machine Learning for the Arts
UCSD SPRING 2019
FINAL PROJECT

ICONOGRAPHIC NEWS

Visualizing AI-related headlines through SketchRNN


SentinelOne raises $120M for its fully-autonomous, AI-based endpoint security solution

Amazon will use AI to help you shop for clothes with StyleSnap

Luke Wulf

DESCRIPTION

Concept: As the world gets more technologically advanced, I see the term of Artificial Intelligence (AI) becoming increasingly personified. However, when one speaks of an intelligence, it implies that the intelligent thing has some form of perspective on its environment. And, with the unique case of AI, it can become hard to realize this perspective due to the virtual nature of the intelligence. Therefore, in this project I attempted to visualize how an AI would interpret headlines of itself by having it first process what a headline was about, and then having it map this meaning into the form of a drawing.

Technique: The pipeline for this project can be split into four separate sections being:

1. Retrieving News Headlines
2. Processing Meaning From Headlines
3. Visualizing Meaning Digitally
4. Visualizing Digital Realization Physically

1. **Headline Retrieval:**
    a. Headline retrieval was done using a python api called NewsAPI[1] , with the API I could obtain the newest headlines, of which I cleaned to only include those that mentioned the term ‘AI’ or ‘Artificial Intelligence’
2. **Processing Headline Meaning:**
    a. Obtaining meaning from the headlines was done in python using the NLTK library. I obtained the different classes of pre-trained sketch models from ML5[2], created wordnets for each, and compared these wordnets with those from the words in the article titles.
    b. I then took the three most similar sketchRNN classes as compared to the title using NLTK’s path_similarity function and exported them to web client.
3. **Visualizing Digitally:**
    a. I visualized the headlines using the JavaScript versions of SketchRNN (ml5.sketchRNN) and Processing[3] (p5). By taking the class outputs from step 2, I could query sketchRNN to obtain a machine-sketched version of the class, which was then drawn onto the client using p5.
4. **Visualizing Physically:**
    a. Next, I exported some of the digital generative sketches as bitmaps, which were then fed into a pen-plotter. After assembling and debugging the pen-plotter, it could accurately draw the digital image.

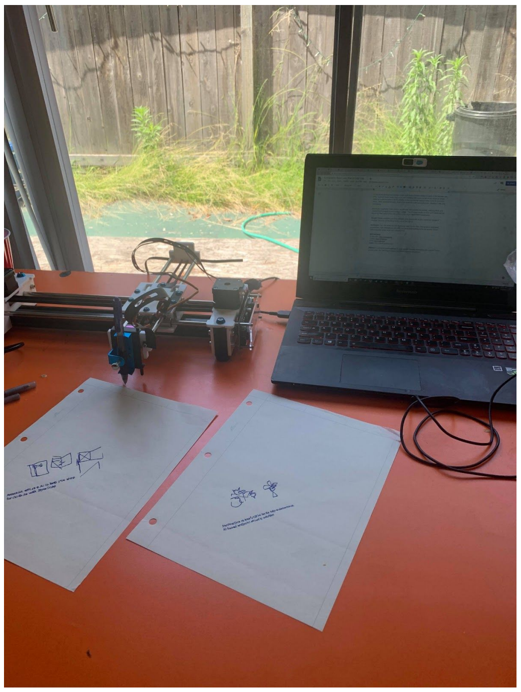
Process: I would describe the process of this project as being very top down. Having built the pen plotter the week before, I wanted to use it io realize one of my ece188 projects. With this being the case, and the restrictions that come with a pen plotter. I chose to stick to the sketch medium since it translates the easiest to an .svg or .bmp that the pen plotter can visualize.

With the medium set, I brainstormed interesting ideas on what to visualize in a generated sketch. At this point I wanted to do something real-time that could be ported eventually to the pen plotter, such as the webcam portrait drawing robot we saw in class. I settled on the relationship of an AI trying to draw itself due to this inspiration.

In the actual development process there were a few troubles related to each of the libraries and APIs, however these mostly had to do with cleaning data since not all words inside an article relate to a valid wordnet in NLTK and not all the pre-trained classes relate to NLTK wordnets either. After cleaning the data, most of the similarity code was python list operations and data processing.

Moving to JavaScript, there were a few hiccups with loading the models, querying ML5, and visualizing the output in one coherent matter. This involved mostly p5 fixes and hacks that led me to my final output of the title in the corner of the sketchRNN visualization.

Result: In terms of the digital result there were a few hiccups since there was a substantial loss of information converting the corpus of words that were in the titles of each article into the ~100 sketchRNN models, therefore keywords such as AI were converted into images such as 'angel', 'book', and 'cellphone', this produced the realization that currently AI can only learn and optimize things that people have exposed them to. In terms of the physical result, the final result turned out pretty good since the pen plotter could accurately convert the digital image into a physical representation using a bitmap format. My workstation and resulting sketches can be seen below in the image. The only issues in this transformation were tradeoffs in speed vs precision, since if you wanted to render the digital image larger or more precisely that would scale the execution time quadratically, therefore a 20cm x 20cm view with 4 lines per mm was chosen to render the generated sketches.



Reflection: All in all, I was satisfied with the results of the project since it was able to produce what I believe to be the best output given the fixed constraints on the model and training data. I was excited to see the NewsAPI and NLTK techniques going to good use, and the semantic similarity functions within NLTK can be very useful for mapping words or phrases into more usable expressions. With regards to the output, I wish it was a bit more linked to the headline in terms of iconographs, however limitations in the model and a lack of hand-drawn data denoting words such as 'AI', 'security', etc led the sketch to configure itself with more manageable classes such as 'book' and 'cellphone'. A future direction may be to link this in real time with a headline listener to draw headlines and articles as they are produced. This would be something like a news ticker. With regards to this course and project, I learned a great deal about how useful machine learning and AI can be in regards to creative tasks, however they are limited still, like many other AI's, by training data and labeling of that data.

REFERENCE:

[1] NewsAPI: https://newsapi.org
[2] ML5: https://ml5js.org/
[3] P5: https://p5js.org/

CODE: https://github.com/ucsd-ml-arts/ml-art-final2-upgamers

RESULT: https://github.com/ucsd-ml-arts/ml-art-final2-upgamers