Gil Pasternak
Elizaveta Pertseva

H.P. Lovecraft Text Generator Project Proposal

Question:
Can a generative deep learning model achieve 90% Turing optimality in a text conversation as H.P. Lovecraft?

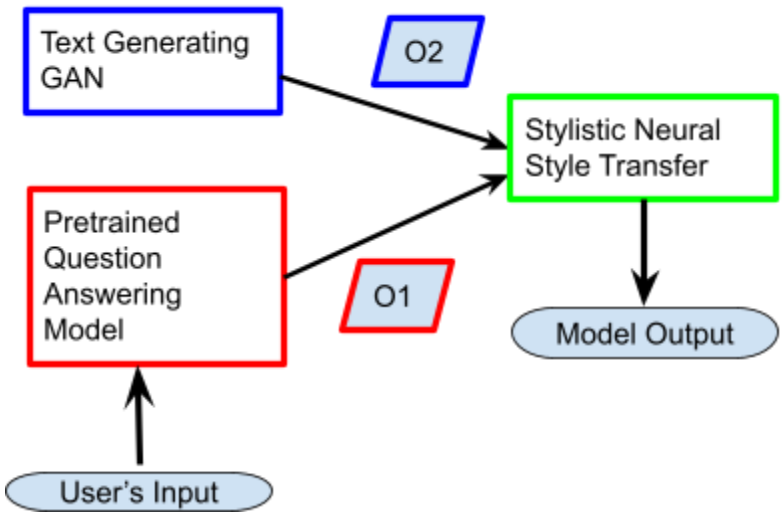
Hypothesis:
A Generative Adversarial Network (GAN) trained on Lovecraft text to learn stylistic representation, then composed with neural style transfer from a pre-trained question answering model and feedforward neural network to merge the two could successfully complete this task with proper hyperparameter tuning.

We do not have data for what a Lovecraft text should look like and the task leaves a lot of room for creativity. However, we believe that by substituting a question answering model we can generate which are reasonable, responsive, and rule-following (grammatical).

Altogether, we believe that we should be able to dillude a human being with previous knowledge of H.P Lovecraft's work into believing the validity (acknowledging the authors writing style and simultaneously remaining unaware of its modeled origin) of the text about 90% of the time.

Methodology + Tools:

We propose using 3 models, one pretrained and 2 built by us in order to find a solution to our question. First we will create and train a GAN on Lovecraft's short stories that will generate new Lovecrat text. Then we will feed the output into an unsupervised learning model that will merge it with a response from a pre trained question answering model.



Assumptions:

- Lovecraft would not use common texting language
- A Question Answering Model is a suitable alternative for a text generating model as it provides context.
- Lovecraft would continue the conversation by asking a relatively normal question, which we may generate from a template (TBD)
- Metric of success is believability

Data:

- **Collection:** :
    1) Initial: Use corpus from github: https://github.com/vilmibm/lovecraftcorpus
    2) Secondary: Web scrape Lovecraft website: https://www.hplovecraft.com/writings/texts/
- **Storage:**
    1) Initially Pandas DataFrames
    2) later transition to RedShift for Data Storage

Deployment:

- A command line interface built initially, eventually serves predictions as a web service with a basic JavaScript UI.

Potential Roadblocks / Hurdles:

- Style Transfer is a not a well researched task when it comes to text
- Style transfer is unsupervised, success is subjective and we will need to develop our own precise metrics
- Hyperparameter tuning may be very extensive given number of neural architectures
- The responses generated by the GAN are so far away from the answer they cannot be converted
- Time
- Scope of project in becoming full stack (should ideally be containerized, optimized, etc.)
- Getting out a MVP before school starts September 22 while both working full time jobs