Assignment 2 - Reflection

The coding segment of the assignment proved tricky in the sense that there are many ways to approach it. With the inclusion of OpenAI for public usage, it became relatively simple, but being new to Python, I felt somewhat limited. As a result, I feel that I haven't done my best work. To make up for this, I tried to emphasize cleanliness, simplicity, and precision.

The code begins by loading the models, then generating 1,000 variations of the poem. I realize that this number is quite high, especially since I was only expected to choose one, so I made it easier on myself by adding a cell that generates and displays 10 random versions to choose from. I also added a debug segment for the first line to ensure the code was working. Finally, the last portion of the code saves the chosen variation of the poem as a text file by altering the x value. In my case, I used it to generate and save P+7 and P+569. I chose P+569 because the first line seemed the wittiest off the bat.

Originally, one idea I had for implementing a P+X value system was to write a code that would generate variations of the poem and then choose one based on a “funniness” scale by asking a prompt like “Which text is the funniest?” I attempted this, but I later learned that this was impossible due to GPT-2’s language model lacking integration for this kind of evaluation, so I was forced to play the role of judge in the end.

One fun idea that crossed my mind for integrating a P+7 technique involves hosting a game show similar to *Family Feud*, in which contestants are asked to name as many words or phrases belonging to a certain category. By using the P+7 technique, the hosts can determine which elements the contestants are most likely to think of. In doing so, they can evaluate which words or phrases would earn them the most points.