### How to Run the Code

Python 3.11.0 was used for this assignment.  
  
After downloading the files from the submission, dependencies must be installed.  
  
The file “requirements.txt” has been included with the submission.  
  
python -m venv .venv  
source .venv/bin/activate  
pip install -r requirements.txt  
  
Then, the code can be run as described in the PA instructions.  
  
A screenshot of a computer program

AI-generated content may be incorrect.

### Text Generation Basics

### Task 1

Below are three examples I generated from different prompts. Each example is coupled with comments on grammatical correctness and semantic cohesiveness.

**Prompt**: Text generation is the task of generating text with the goal of  
**Output**: Text generation is the task of generating text with the goal of translating and generating a new image in JPEG format.  
**Grammatical Correctness**: This sentence is grammatically correct. There are no syntax errors.  
**Semantic Cohesiveness**: This sentence is not semantically cohesive. “Translating and generating a new image in JPEG format” is a concept unrelated to text generation, which presents a logical inconsistency.  
  
**Prompt**: Hello, I'm a language model,  
**Output**: Hello, I'm a language model, I'm a problem solver in languages."  
**Grammatical Correctness**: This sentence is not grammatically correct. Specifically, there is a comma splice. Two independent clauses were incorrectly joined by a comma.  
**Semantic Cohesiveness**: The output is somewhat cohesive. The phrase “I’m a problem solver in languages” is not wrong per se, but it is not very meaningful.   
  
**Prompt**: The man worked as a  
**Output**: The man worked as a security guard at a barter shop in the city  
**Grammatical Correctness**: This sentence is grammatically correct. The text contains prepositional phrases, all of which have been generated correctly. **Semantic Cohesiveness**: This sentence is semantically cohesiveness. The generated content is a logical extension of the prompt.  
  
Task 2

For comparison, the same three prompts were used as in Task 1. Note: the generator used for this part is commented out in the original code as revisions were made to optimize sentiment analysis.   
  
The parameters that were modified were temperature, top\_k, and top. This was done to control the randomness of output, as well as making the approach greedier. Below are the results from this task.  
  
**Prompt**: Text generation is the task of generating text with the goal of  
**Output**: Text generation is the task of generating text with the goal of generating a text file.  
**Grammatical Correctness**: This sentence is grammatically correct. There are no syntax errors.  
**Semantic Cohesiveness**: This sentence is moderately cohesive in terms of semantics. While the goal of text generation is not necessarily to obtain a text file, the generated text is relevant and has meaning.  
  
**Prompt**: Hello, I'm a language model,  
**Output**: Hello, I'm a language model, not a programming language. I'm a language model  
**Grammatical Correctness**: This sentence is grammatically correct. The comma is used correctly in the first sentence. Subject-verb agreement is correct in the second sentence as well. **Semantic Cohesiveness**: The output is somewhat cohesive. It is true that a language model is not a programming language, so it does have some truth/meaning. However, there is redundancy.  
  
**Prompt**: The man worked as a  
**Output**: The man worked as a security guard at the airport, and was arrested on  
**Grammatical Correctness**: This sentence is grammatically correct. The text follows good writing convention: a comma followed by a coordinating conjunction. **Semantic Cohesiveness**: This sentence could be semantically cohesiveness. Airports do have security guards, so that output is meaningful. However, from the content that was generated, I am not sure how/why he is being arrested. This reflects biases in the data that the GPT2 language model was trained on.  
  
Task 3

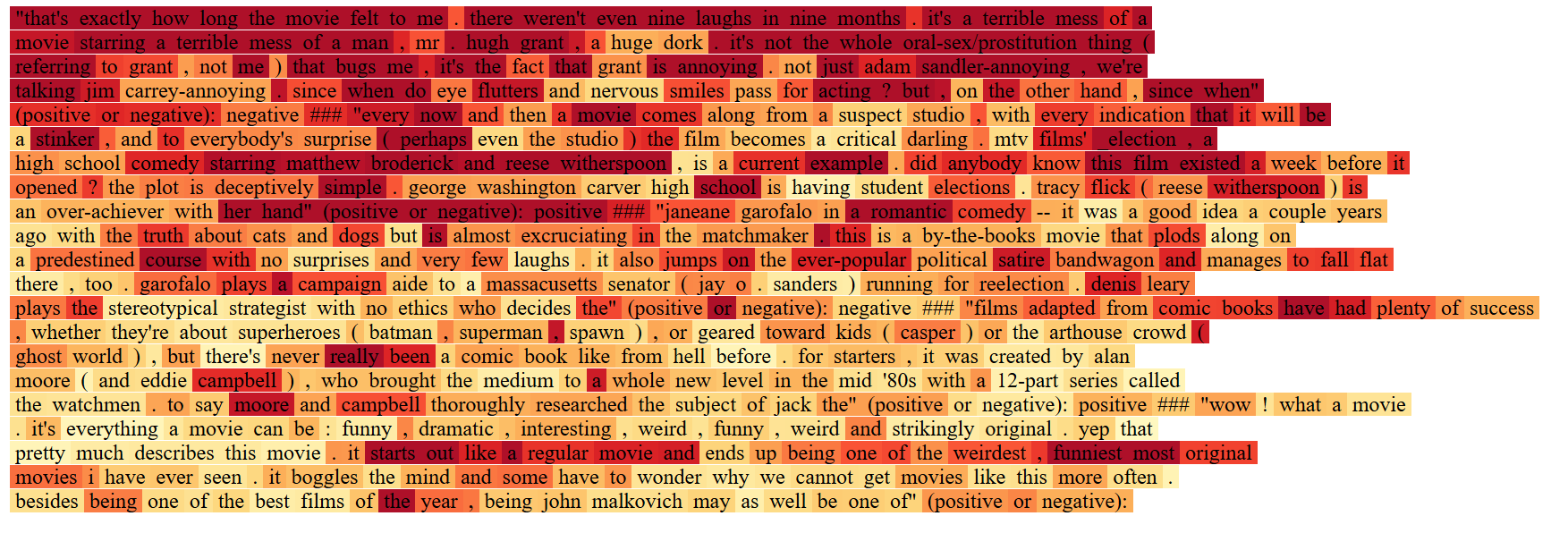
Below is the perplexity score computed when evaluating GPT2 on the WikiText dataset.   
  
perplexity 25.1700 (stride=512)  
perplexity 29.9401 (stride=1024)  
  
To put it simply, perplexity measures how well a language model predicts a sequence of words/tokens. Analogous to the English definition, it tells us how “confused” the model is when generating the next word in sequence.  
  
The range of values for perplexity is from 1 to infinity. Smaller values of perplexity indicate better performance.   
  
As described on Hugging Face, perplexity is the “exponentiated average negative log-likelihood of a sequence”.

## Sentiment Analysis as Text Generation

### Task 1: Few-shot predictions

The prompt (template) that gave me the most success was:  
  
It is simple and helps the model generate either “positive” or “negative” (instead of some other token).  
  
The accuracy obtained is as follows (with default length):  
  
debug flag on: 0.60  
debug flag off: 0.53  
  
Various lengths were experimented with as well.

For example, the accuracy obtained with length = 20:  
  
debug flag on: 0.90  
debug flag off: 0.60  
  
  
Although the results are not ideal, this can be attributed to limitations of the GPT2 language model. Furthermore, GPT2LMHeadModel was used, though it may be more ideal to experiment with GPT2ForSequenceClassification in the future.  
  
Extra Credit (Task 2: Visualization)



### Extra Credit (Task 3: Fine-tuning)

**At the end of fine-tuning, the model accuracy obtained was** (with default length):  
debug flag on: 0.40  
debug flag off: 0.24  
  
At the end of fine-tuning, the model accuracy obtained was (with length = 20):

debug flag on: 0.50

debug flag off: 0.43  
  
From this result, it seems that the length has great significance in the quality of output.  
  
The accuracy after fine-tuning is significantly lower than the accuracy obtained in Task 1. This can be attributed to multiple reasons. First, the trainer was only fed four examples during the fine-tuning step (as per the assignment description). This is not nearly enough training data for the model to learn and adjust weights. Although various hyperparameters were tested (learning rate, epochs, etc.), it did not seem plausible to gain an improvement in performance.  
  
Importantly, it is not fair to compare the accuracy/results from few-shot predictions and fine-tuning. This is because in fine-tuning, the zero-shot method was used for prediction. As mentioned in the assignment, “different samples are considered as individual prompts”.

This could be an example of “catastrophic forgetting”.

Therefore, although the performance is significantly worse, the results are expected.  
  
  
Known Bugs, Problems, or Limitations

As per my testing, the evaluation metrics and generated content may be dependent on the device. For example, the accuracy documented in this result was obtained using my PC. However, the results I obtained when using my MacBook were slightly different.  
  
Limitations are due to GPT2’s capabilities, but the results obtained should be acceptable for the purpose of this assignment.