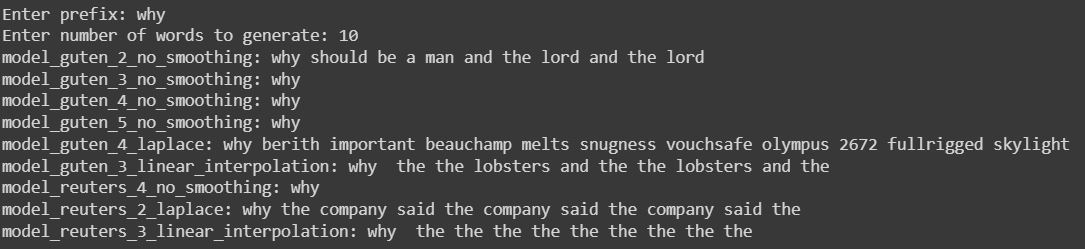
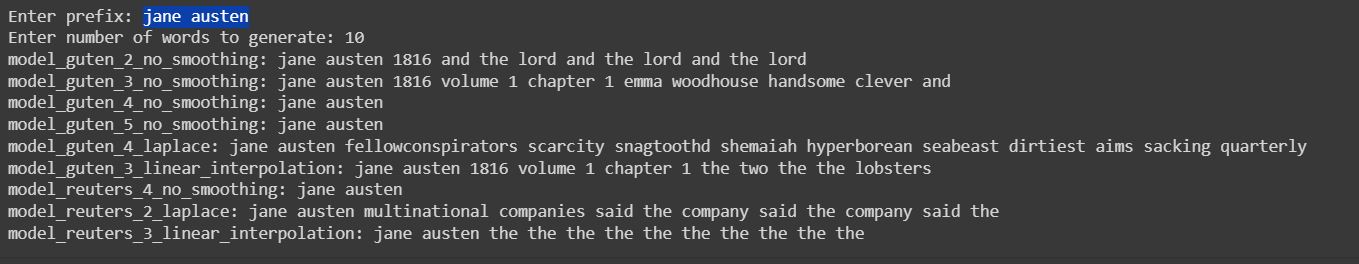
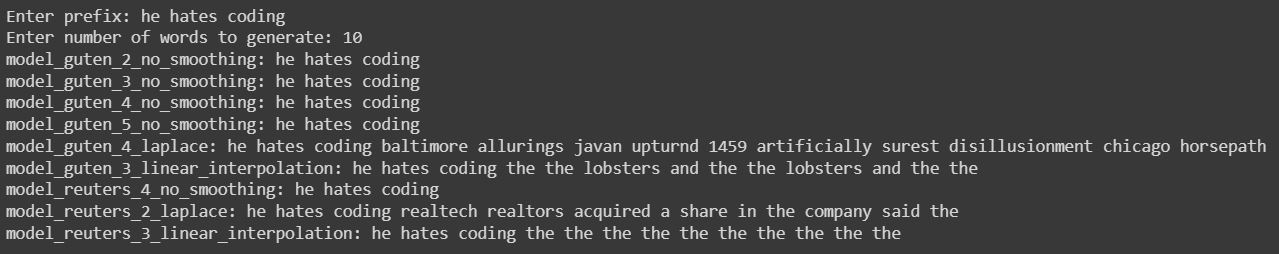
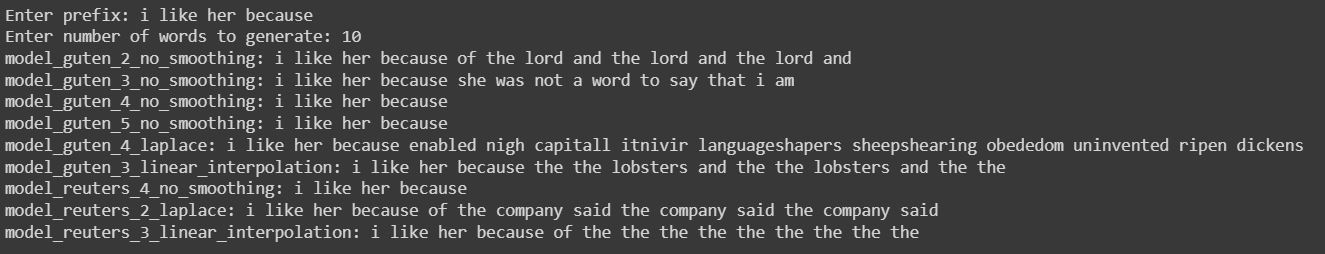
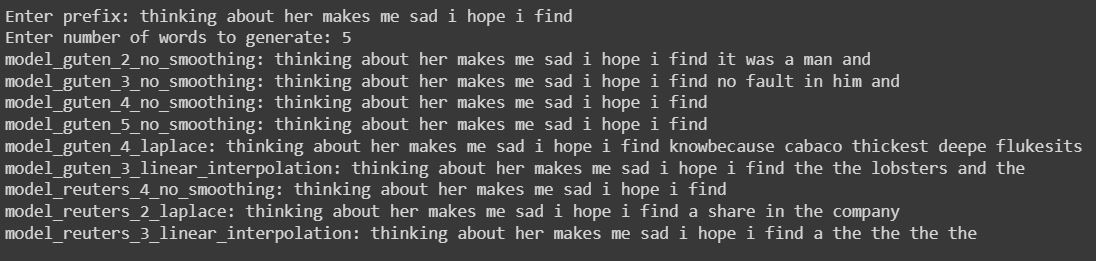
Akshay Syal

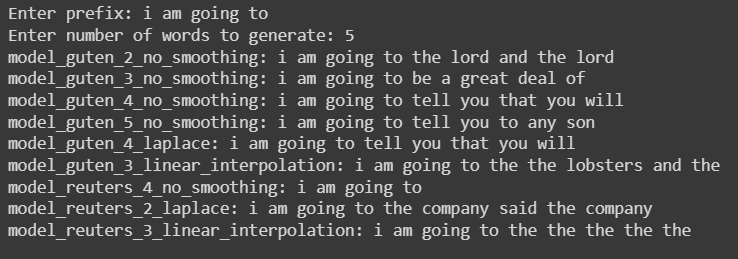
\* I have discussed my solution and code with Rishab Jaiprakash Khuba

3. Testing and evaluation:

(a) Test the model with different (n - 1)-gram inputs to see how it predicts the next word.

(b) Compare the performance of different n values (e.g., bigrams vs. trigrams).





4. Write a short report discussing your results.

* Bigram and Trigram models often run into a loop for eg. Generating words like ‘and the lord’ over and over
* Laplace smoothing and linear interpolation can generate sentences even if the word in the prompt is not in the dataset on which the model is compiled.
* If the n-grams of the prompt are in dataset then the output of laplace and no-smoothing model is same
* When smoothing is linear interpolation then if the model is prompted by a word which is not in vocabulary, then the most common word is returned. This is because linear interpolation calculates probabilities for trigrams, bigrams and unigrams. Probability of unigram ‘the’ is highest as ‘the’ is the most common word in English.