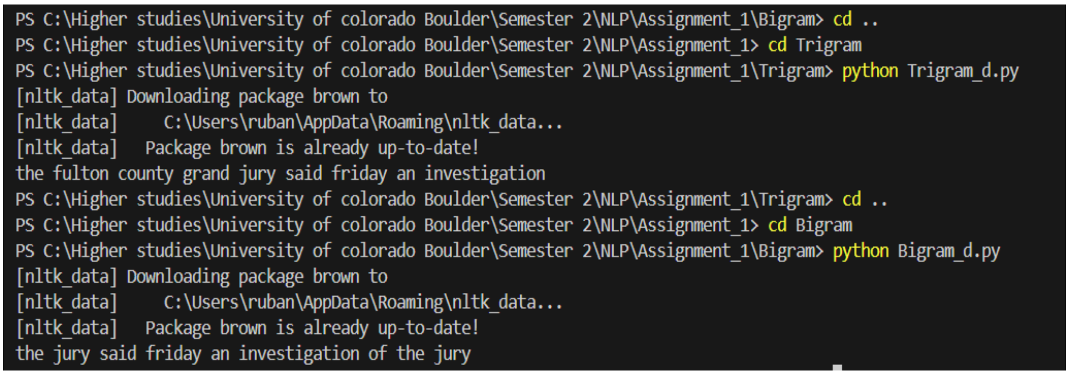
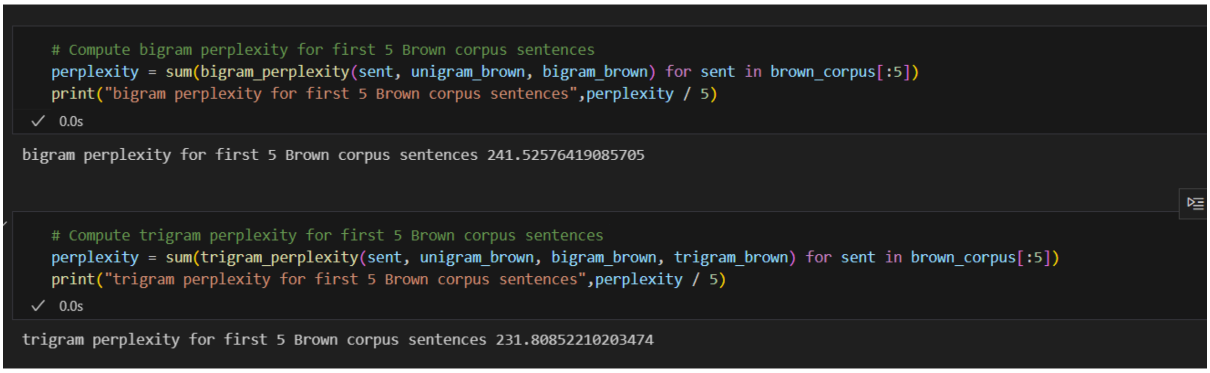
1) a) The key difference between the bigram and trigram models lies in the context they use for predictions. A bigram model considers only the previous word to predict the next word, whereas a trigram model takes the two preceding words to predict the next word. Because of this, the trigram model generally produces more coherent text, as it has a broader context for making predictions. The below screenshot shows the predict of sentence using bigram and trigram using brown corpus. To Run this code, inside the Bigram folder, run the Bigram_d.py file and inside the Trigram folder, run the Trigram_d.py file



```
PS C:\Higher studies\University of colorado Boulder\Semester 2\NLP\Assignment_1\Bigram> cd ..
PS C:\Higher studies\University of colorado Boulder\Semester 2\NLP\Assignment_1> cd Trigram
PS C:\Higher studies\University of colorado Boulder\Semester 2\NLP\Assignment_1\Trigram> python Trigram_d.py
[nltk_data] Downloading package brown to
[nltk_data]     C:\Users\ruban\AppData\Roaming\nltk_data...
[nltk_data]   Package brown is already up-to-date!
the fulton county grand jury said friday an investigation
PS C:\Higher studies\University of colorado Boulder\Semester 2\NLP\Assignment_1\Trigram> cd ..
PS C:\Higher studies\University of colorado Boulder\Semester 2\NLP\Assignment_1> cd Bigram
PS C:\Higher studies\University of colorado Boulder\Semester 2\NLP\Assignment_1\Bigram> python Bigram_d.py
[nltk_data] Downloading package brown to
[nltk_data]     C:\Users\ruban\AppData\Roaming\nltk_data...
[nltk_data]   Package brown is already up-to-date!
the jury said friday an investigation of the jury
```

b) In terms of perplexity, the trigram model is expected to have a lower perplexity than the bigram model. A lower perplexity indicates a better model since it assigns higher probabilities to the correct word sequences. However, if the training dataset is small, the trigram model might suffer from data sparsity, which can lead to higher perplexity than expected.

c) To test this, I trained models on the Brown corpus and evaluated their perplexity on 2-5 sentences from the brown corpus. The results align with my expectations, showing that the trigram model generally performs better in terms of coherence and perplexity. I have attached a screenshot of the results.



```
# Compute bigram perplexity for first 5 Brown corpus sentences
perplexity = sum(bigram_perplexity(sent, unigram_brown, bigram_brown) for sent in brown_corpus[:5])
print("bigram perplexity for first 5 Brown corpus sentences",perplexity / 5)
```

✓ 0.0s

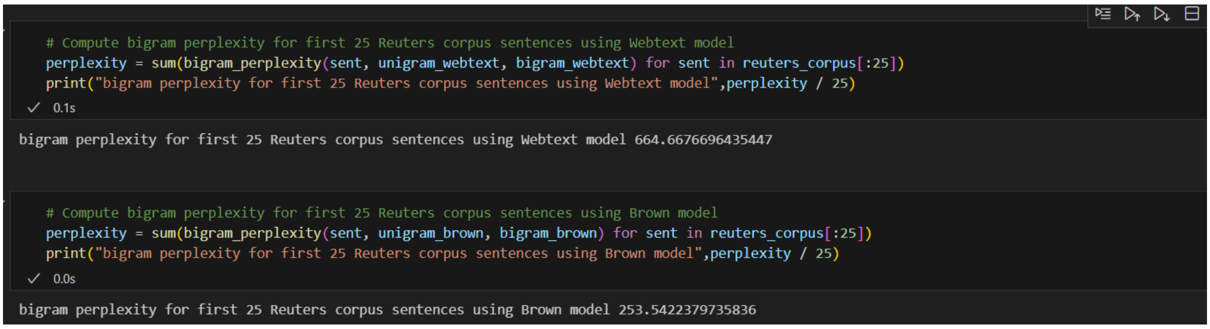
```
bigram perplexity for first 5 Brown corpus sentences 241.52576419085705
```

```
# Compute trigram perplexity for first 5 Brown corpus sentences
perplexity = sum(trigram_perplexity(sent, unigram_brown, bigram_brown, trigram_brown) for sent in brown_corpus[:5])
print("trigram perplexity for first 5 Brown corpus sentences",perplexity / 5)
```

✓ 0.0s

```
trigram perplexity for first 5 Brown corpus sentences 231.80852210203474
```

2) The bigram model trained on the Brown corpus performed better. The average perplexity for the Brown-based bigram model on Reuters was 253.54, while the Webtext-based bigram model had an average perplexity of 664.66. I believe this result is because the Reuters corpus consists largely of domain-specific and news-related vocabulary. The Brown corpus also contains more formal sentences, which aligns more closely with the style of language used in Reuters articles. In contrast, Web text has more informal language, which is less similar to the vocabulary of Reuters. As a result, the Brown-based model performs better when tested on the Reuters data.



```
# Compute bigram perplexity for first 25 Reuters corpus sentences using Webtext model
perplexity = sum(bigram_perplexity(sent, unigram_webtext, bigram_webtext) for sent in reuters_corpus[:25])
print("bigram perplexity for first 25 Reuters corpus sentences using Webtext model",perplexity / 25)
```

✓ 0.1s

```
bigram perplexity for first 25 Reuters corpus sentences using Webtext model 664.6676696435447
```

```
# Compute bigram perplexity for first 25 Reuters corpus sentences using Brown model
perplexity = sum(bigram_perplexity(sent, unigram_brown, bigram_brown) for sent in reuters_corpus[:25])
print("bigram perplexity for first 25 Reuters corpus sentences using Brown model",perplexity / 25)
```

✓ 0.0s

```
bigram perplexity for first 25 Reuters corpus sentences using Brown model 253.5422379735836
```

3) Increasing the number of sentences in training data can help improve the model's ability to predict the next word more accurately. With more data, the model can learn better patterns, and relationships between words, leading to more accurate predictions.