Calculating Sentence Probability using a 2-gram Model

Here’s implementing Bi-gram language model using the NLTK package and the Brown corpus. The process is as follows:

1. **User Input**:
   - The code prompts the user to enter a sentence from the keyboard.
   - The entered sentence is converted to lowercase to maintain consistency in processing.
2. **Building the Language Model**:
   - I utilize NLTK's bigrams function to generate bigrams from the user-entered sentence, including special tokens <s> (start of sentence) and </s> (end of sentence) to handle sentence boundaries.
   - Each bigram starting or ending a sentence is assumed to have a probability of 0.25, as specified.
   - The probability of internal bigrams is calculated using frequency counts from the Brown corpus.
3. **Probability Calculation**:
   - The total probability of the sentence is computed by multiplying the probabilities of all bigrams in the sentence.
   - If a bigram is not found in the training data, the sentence probability defaults to zero.
4. **Output**:
   - The program displays the processed sentence, the list of bigrams, their individual probabilities, and the final probability of the entire sentence.

Results:

Enter Sentence (read in ‘S’) : Good Morning

Applying lowercasing to S: good morning

Here, I took **Bigrams by sentence**

Output Displayed:

Displayed sentence: Entered Sentence is : 'good morning'

List of all individual bigrams: Possible bigrams for above entered sentence are :

[('<s>', 'good'), ('good', 'morning'), ('morning', '</s>')]

Their probabilities:

probability of ('<s>', 'good'): 0.25

probability of ('good', 'morning'): 0.0012406947890818859

probability of ('morning', '</s>'): 0.25

Final probability of sentence:

The final probability of the sentence '<s> good morning </s>' is approximately 7.754342431761787e-05