Experiment No.5
Implement Bi-Gram model for the given Text input
Date of Performance:
Date of Submission:

Aim: Implement Bi-Gram model for the given Text input

Objective: To study and implement N-gram Language Model.

Theory:

A language model supports predicting the completion of a sentence.
Eg:

- Please turn off your cell _____
- Your program does not _____

Predictive text input systems can guess what you are typing and give choices on how to complete it.

N-gram Models:
Estimate probability of each word given prior context.
P(phone | Please turn off your cell)

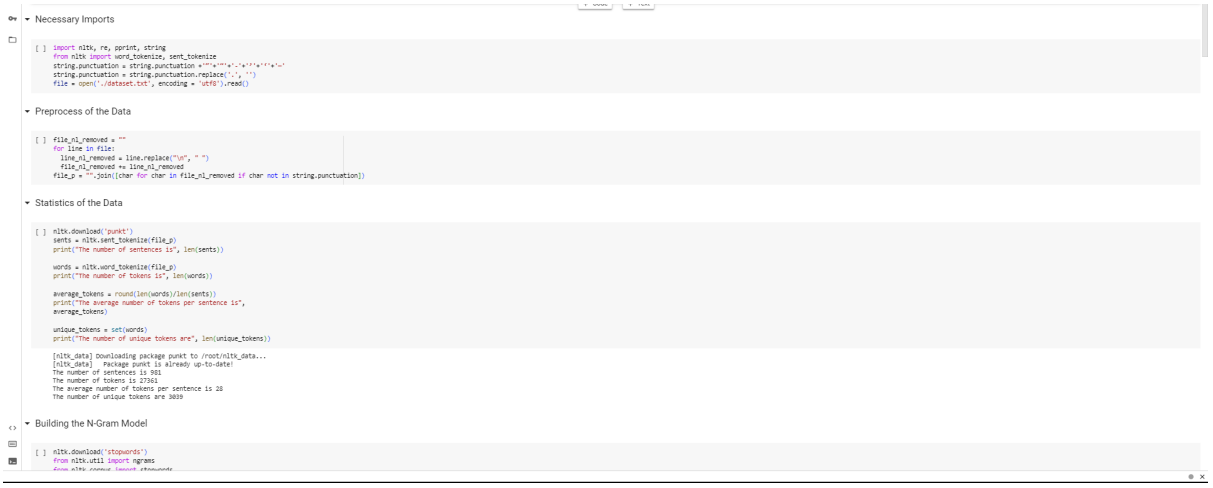
- Number of parameters required grows exponentially with the number of words of prior context.
- An N-gram model uses only N1 words of prior context.
    - Unigram: P(phone)
    - Bigram: P(phone | cell)
    - Trigram: P(phone | your cell)

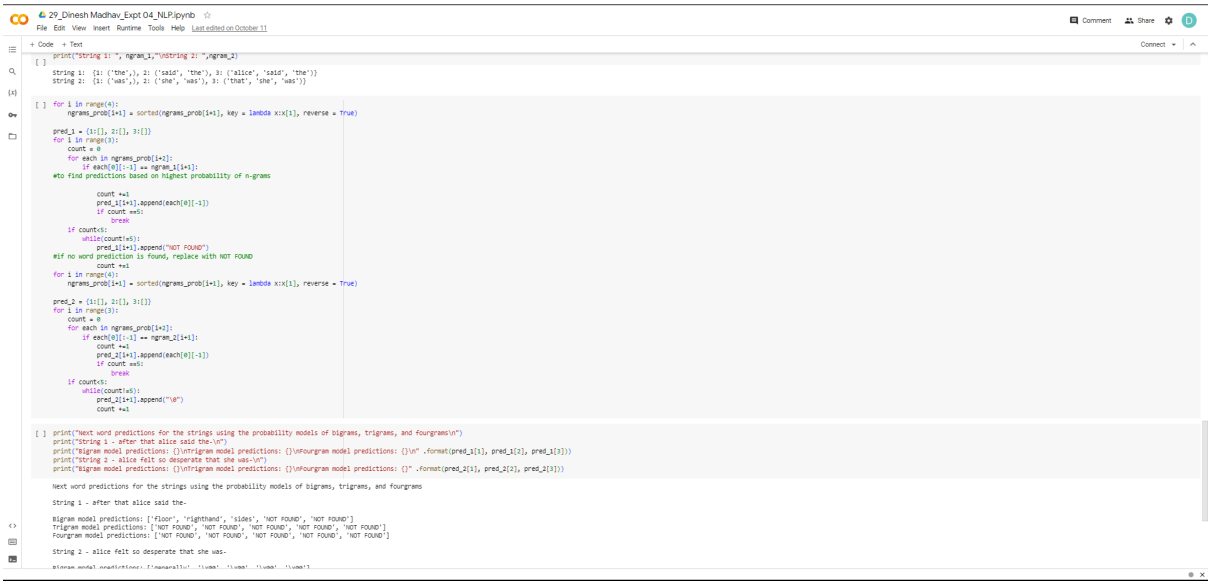
- The Markov assumption is the presumption that the future behavior of a dynamical system only depends on its recent history. In particular, in a kth-order Markov model, the next state only depends on the k most recent states, therefore an N-gram model is a (N1)-order Markov model.

N-grams: a contiguous sequence of n tokens from a given piece of text

Fig. Example of Trigrams in a sentence

Output:





Conclusion:

N-gram language models are statistical models that predict the next word in a sequence based on the previous N-1 words. They are often used in NLP tasks such as speech recognition, machine translation, and text generation.

The results of N-gram language models depend on the size and quality of the training corpus, the order of the N-gram model, and the smoothing algorithm used.

In general, N-gram language models are effective in a variety of NLP tasks, but they can be computationally expensive to train and use, and they may not perform well on data that is different from the training corpus.