LANGUAGE MODELING

Introduction :

This assignment is similar to the first assignment of NLP Course where we developed a tweet tokenizer and outputted each token on a new line for every input tweet. In this assignment, we extend the tweet tokenizer to n-gram model where n tokens form one token. The model can be bi-gram (2 tokens together), tri-gram (3 tokens together), 4-gram (4 tokens together) or 5-gram.

Logic:

1) This assignment is completed using first assignment of tweet tokenizer with some modifications in the output.
2) The output from first assignment consists of list of individual tokens. Instead of printing each token on new line, the program prints consecutive 2, 3, 4 or 5 tokens on one line and then increment the index of list by 1 again print the nest 2, 3, 4 or 5 tokens on next line and so on.
3) For bi-gram language model, the program prints 2 consecutive tokens from the list on the same line and separated by space.

For eg: woman who missed => woman who
who missed
2 consecutive words form one token in bi-gram model.

4) For tri-gram language model, the program prints 3 consecutive tokens from the list of individual tokens, on the same line and separated by space.

For eg:

“Hello!All” => “ Hello !
Hello ! All
! All “
5) For 4-gram language model, the program prints 4 consecutive tokens from the list of individual tokens, on the same line and separated by space.

For eg:

woman who missed flight for Pune => woman who missed flight
who missed flight for
missed flight for Pune

6) For 5-gram language model, the program prints 5 consecutive tokens from the list of individual tokens, on the same line and separated by space.

For eg:

woman who missed flight for Pune => woman who missed flight for
who missed flight for Pune

Datasets used:

1) Fetched famous and most-liked tweets from wikipedia and prepared a dataset.

Resources:

1) Python Documentation of re module
2) Python Documentation of NLTK