FAST National University of

Computer and Emerging Sciences

Natural Langauge Processing

Assignment 1

- Name: Muneel Haider
- Roll No: 21I-0640

Table of Content:

Problem Introduction:

The assignment revolved around developing a NLP system comprising on N-gram models, which are capable of predicting the next word of a sentence based on the CSV file provided. The CSV file consisted of movie reviews, based on which the N-gram models were created. The N-gram models predict words based on the assumption that the word in a text depends only on the previous n-words (this n depends on the type of N-gram model). This principle is applied to generate text and classify movie reviews as either Positive or Negative.

Problem Objective:

1. **Text Prediction:** Build models that can predict the next word of a phrase based on the previous words using Unigram, Bigram and Trigram models.
2. **Review Classification:** Classify movie reviews as Positive or Negative using the generated text from the N-gram models.

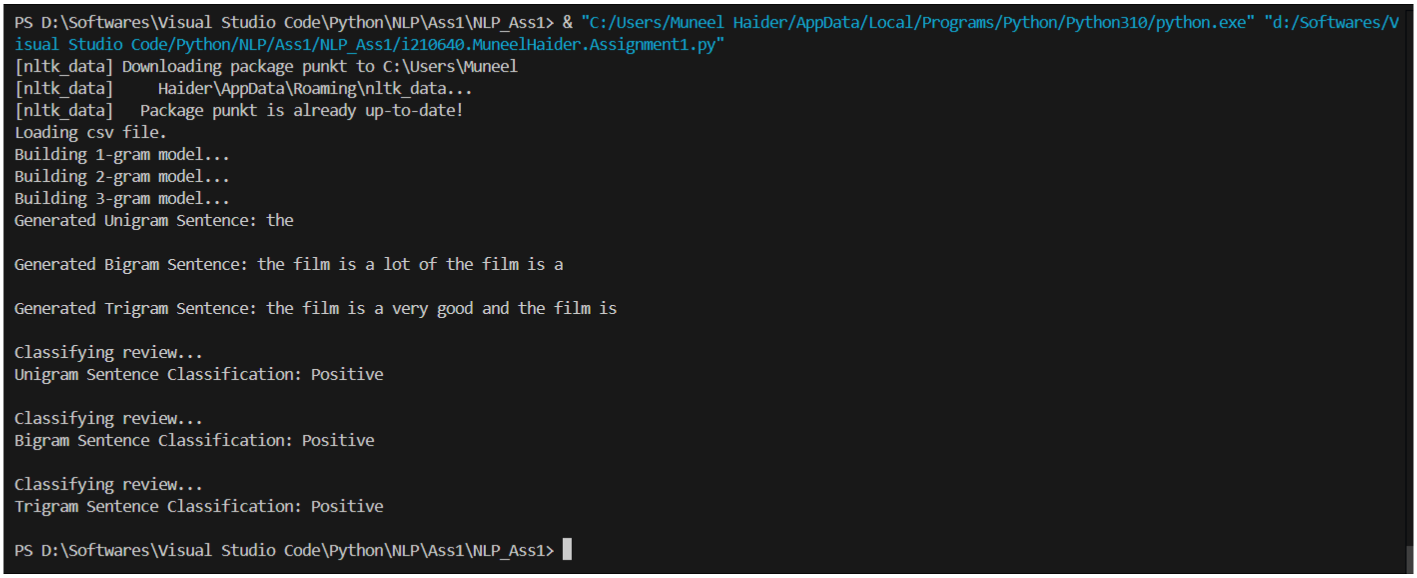
Steps for Solution:

1. **Data Preparation:** Read and preprocess movie reviews data from the CSV file, which includes converting text to lowercase, removing punctuation and <br> tags, and tokenizing.
2. **Model Building:** Build Unigram, Bigram and Trigram models from tokenized text.
3. **Sentence Generation:** Generate sentences using the N-gram models to show the predictive capability of each model.
4. **Review Classification:** Classify these generated sentences as Positive or Negative.

Logic Around the Solution:

- **Building n-gram model:** Each token from the reviews processed text is used to construct n-gram models where the occurrence of each word is counted in the context of the preceding words..
- **Sentence Generation Logic:** The model predicts the next word by choosing the most frequent adjacent word from the model.
- **Classification Logic:** The classification of reviews is based on the sum of the Positive and Negative weights of the words in the generated sentences. Words present in the model contribute positively, while absent words contribute negatively.

Output:



```
PS D:\Softwares\Visual Studio Code\Python\NLP\Ass1\NLP_Ass1> & "C:/Users/Muneel Haider/AppData/Local/Programs/Python/Python310/python.exe" "d:/Softwares/V
isual Studio Code/Python/NLP/Ass1/NLP_Ass1/i210640.MuneelHaider.Assignment1.py"
[nltk_data] Downloading package punkt to C:\Users\Muneel
[nltk_data]     Haider\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
Loading csv file.
Building 1-gram model...
Building 2-gram model...
Building 3-gram model...
Generated Unigram Sentence: the

Generated Bigram Sentence: the film is a lot of the film is a

Generated Trigram Sentence: the film is a very good and the film is

Classifying review...
Unigram Sentence Classification: Positive

Classifying review...
Bigram Sentence Classification: Positive

Classifying review...
Trigram Sentence Classification: Positive

PS D:\Softwares\Visual Studio Code\Python\NLP\Ass1\NLP_Ass1> ▌
```

Conclusion:

The system shows the fundamental capabilities of n-gram models in text prediction and sentiment analysis. While this task provides a basic framework for understanding NLP, improvements such as smoothing techniques and advanced machine learning models could improve both the quality of text generation and accuracy of review classification.