NLP Experiment 6

Aim: To implement Text Processing Models

Theory:

In this assignment, we delve into the practical implementation of two foundational text processing techniques: the N-gram model (including 2-gram and 3-gram) and the Term Frequency- Inverse Document Frequency (TF-IDF) model. These methodologies are integral to natural
language processing (NLP), providing crucial insights into text patterns and enabling tasks such as text prediction and document similarity analysis.

The N-gram model involves calculating probabilities of word sequences, where 2-gram and 3-gram models capture the likelihood of a word given its preceding words. Tokenization and N-gram generation are facilitated using the Natural Language Toolkit (NLTK). The 2-gram model utilizes a defaultdict structure to efficiently capture relationships between prefixes and suffixes.

Enhancements include factoring in word frequencies for more nuanced predictions.

On the other hand, the TF-IDF model focuses on term frequency and inverse document frequency. Scikit-learn is employed for TF-IDF vectorization, providing a robust toolkit for numerical operations. The TF-IDF vectorizer is configured with English stop words for effective preprocessing, ensuring a cleaner and more representative analysis of textual data. Cosine similarity calculation is integrated for a comprehensive measure of document similarity analysis.

Example for N-Gram Model:
Consider the input text "A young scriptwriter." For 2-grams, predictions for 'scriptwriter' include terms like screenwriter, working, and novelist. For 3-grams, predictions include phrases like 'who had just,' 'who is,' and 'who dreams.'

Example for TF-IDF Model:
For the TF-IDF model, take the input text "Three best friends spy on their families, sneak into each other's house, and organize elaborate pranks." This yields top 5 similar documents with corresponding cosine similarity values.

Formula

Bigram: $$P(w_n \mid w_{n-1}) = \frac{C(w_{n-1} w_n)}{C(w_{n-1})}$$

N-gram: $$P(w_n \mid w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1} w_n)}{C(w_{n-N+1}^{n-1})}$$

$$TF(t,d) = \frac{number\ of\ times\ t\ appears\ in\ d}{total\ number\ of\ terms\ in\ d}$$

$$IDF(t) = log\frac{N}{1+df}$$

$$TF - IDF(t,d) = TF(t,d) * IDF(t)$$

Libraries and Tools Used:

- NLTK (Natural Language Toolkit).
- Collections.Counter
- Math
- Pandas