Natural Language Processing
Assignment 1

Name: Harsha K

Corpus used: https://www.gutenberg.org/ebooks/75350 (Plain text utf)

Design Approach

The implementation follows an n-gram language modeling approach to analyze text, predict words, and evaluate the model's perplexity.

The approach is structured as follows:

1. Preprocessing Text
   - Converts text to lowercase.
   - Removes non-alphanumeric characters.
   - Tokenizes sentences and words.
   - Removes stopwords and applies lemmatization.
2. Building the N-Gram Model
   - Extracts n-grams from tokenized text.
   - Computes probabilities based on frequency.
3. Calculating Perplexity
   - Evaluates the model's performance on test data.
   - Applies smoothing for unseen n-grams.
4. Predicting the Next Word
   - Uses the trained model to predict the next word given (n-1) previous words.
5. Loading and Processing Corpus
   - Reads text data from a file and processes it.
6. Building Unigram, Bigram, and Trigram Models
   - Constructs models for different n-gram sizes.
7. Example Prediction & Perplexity Calculation
   - Demonstrates word prediction and perplexity calculation on a sample input.

Solution

The solution is implemented in the preprocess_text() function, which takes a string of text as input and returns a list of preprocessed tokens.

The function performs the following steps:

Convert to Lowercase: The input text is converted to lowercase using the lower() method.
Remove Non-Alphanumeric Characters: The re.sub() function is used to remove any character that is not a letter or number.
Sentence Tokenization: The sent_tokenize() function from NLTK is used to split the text into sentences.
Word Tokenization: The word_tokenize() function from NLTK is used to split each sentence into words.
Flattening the List of Tokens: The list of lists of tokens is flattened into a single list.

Stopword Removal: The tokens are filtered to remove any stopwords using NLTK's list of English stopwords.
Lemmatization: The tokens would be lemmatized using NLTK's WordNetLemmatizer.

Used Probability & Perplexity to evaluate the model.

Perplexity Value:
Query from the corpus used: “The exclusion of women from trades is in most cases”
Unigram model next word prediction: No prediction available
Bigram model next word prediction: desertion
Trigram model next word prediction: notoriously
Unigram Perplexity: 542.866253690762
Bigram Perplexity: 3821.3405443311485
Trigram Perplexity: 6065.0

Conclusion

This solution provides a lightweight n-gram-based NLP approach suitable for text modeling and prediction. It efficiently processes a given corpus, predicts next words, and evaluates perplexity, making it applicable to basic NLP tasks, autocomplete systems, and language modeling studies.