Report for Task 2: Learning Word Embeddings with CBOW

Objective:
The objective of this task was to train a Continuous Bag of Words (CBOW) model using the Word2Vec algorithm on the 20 Newsgroups dataset. The goal was to generate word embeddings and subsequently create document embeddings by averaging the word embeddings of the words in each document.

Steps Performed:
Importing Libraries:

Essential libraries such as nltk, gensim, numpy, and sklearn were imported to facilitate text processing, model training, and data handling.

Downloading NLTK Resources:

The necessary NLTK resources (punkt for tokenization and stopwords for removing common words) were downloaded.

Loading the Dataset:

The 20 Newsgroups dataset was loaded with selected categories related to technology, recreation, and science. The dataset was preprocessed by removing headers, footers, and quotes to focus on the main content.

Text Preprocessing:

A preprocessing function was defined to:

Convert text to lowercase.

Tokenize the text into words.

Remove punctuation and numbers.

Eliminate stopwords.

This function was applied to all documents in the dataset to prepare the text for training.

Training the CBOW Model:

The CBOW model was trained using the Word2Vec implementation from the gensim library. The model was configured with:

A vector size of 100.

A window size of 5.

A minimum word count of 2.

The sg=0 parameter to specify CBOW (as opposed to Skip-Gram).

Generating Document Embeddings:

A function was created to compute document embeddings by averaging the word embeddings of all words in a document that are present in the model's vocabulary.

Document embeddings were generated for all documents in the dataset.

Saving the Model and Embeddings:

The trained CBOW model and the generated document embeddings were saved to disk for future use.

Example Word Embeddings:

The word embeddings for the words "computer" and "space" were printed as examples to demonstrate the output of the trained model.

Results:
The CBOW model was successfully trained on the 20 Newsgroups dataset.

Document embeddings were generated by averaging the word embeddings of the words in each document.

The model and embeddings were saved to disk, ensuring that they can be reused without retraining.

Example word embeddings for "computer" and "space" were displayed, showing the 100-dimensional vectors generated by the model.

Conclusion:
The task was completed successfully, and the CBOW model was able to generate meaningful word embeddings for the given dataset.

The document embeddings can be used for various downstream tasks such as document classification, clustering, or similarity analysis.

The model's performance can be further evaluated by testing it on specific NLP tasks or by visualizing the embeddings using techniques like t-SNE or PCA.

Future Work:
Experiment with different hyperparameters (e.g., vector size, window size) to optimize the model's performance.

Compare the CBOW model's performance with the Skip-Gram model on the same dataset.

Use the generated embeddings for specific NLP tasks such as text classification or clustering to evaluate their effectiveness.

Code Execution:
The code executed without errors, and the outputs (word embeddings and document embeddings) were generated as expected.

The model and embeddings were saved successfully, ensuring reproducibility and future use.