In [ ]:

#Extracting the frequency of terms using the Bag of Words model

import numpy as np

from sklearn.feature\_extraction.text import CountVectorizer

from nltk.corpus import brown

from text\_chunker import chunker

# Read the data from the Brown corpus

input\_data = ' '.join(brown.words()[:5400])

# Number of words in each chunk

chunk\_size = 800

#Divide the input text into chunks:

text\_chunks = chunker(input\_data, chunk\_size)

Convert the chunks into dictionary items:

# Convert to dict items

chunks = []

for count, chunk in enumerate(text\_chunks):

d = {'index': count, 'text': chunk}

chunks.append(d)

# Extract the document term matrix

count\_vectorizer = CountVectorizer(min\_df=7, max\_df=20)

document\_term\_matrix = count\_vectorizer.fit\_transform([chunk['text'] for chunk in chunks])

# Extract the vocabulary and display it

vocabulary = np.array(count\_vectorizer.get\_feature\_names())

print("\nVocabulary:\n", vocabulary)

# Generate names for chunks

chunk\_names = []

for i in range(len(text\_chunks)):

chunk\_names.append('Chunk-' + str(i+1))

# Print the document term matrix

print("\nDocument term matrix:")

formatted\_text = '{:>12}' \* (len(chunk\_names) + 1)

print('\n', formatted\_text.format('Word', \*chunk\_names), '\n')

for word, item in zip(vocabulary, document\_term\_matrix.T):

# 'item' is a 'csr\_matrix' data structure

output = [word] + [str(freq) for freq in item.data]

print(formatted\_text.format(\*output))

