Tokenizataion-1 (Score: 100.0 / 100.0)

1. [Test cell](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg#cell-dea8ced7b5063ad6) (Score: 25.0 / 25.0)
2. [Test cell](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg#cell-fecaffd35d0f0438) (Score: 25.0 / 25.0)
3. [Test cell](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg#cell-2b3c0e9cce5e4ba5) (Score: 25.0 / 25.0)
4. [Test cell](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg#cell-8440f3e352a5caf8) (Score: 25.0 / 25.0)

You are tasked with creating a pipeline for preparing textual data, which involves tokenization, lemmatization, stemming, stop-word removal, and embedding word vectors.

1. Tokenization and Lowercasing Break the input text into individual words (tokens) while removing punctuation and whitespace. Convert all tokens to lowercase to maintain uniformity.
2. Lemmatization Simplify the tokens by converting them to their base or dictionary form, ensuring that different grammatical forms of the same word are treated as one.
3. Stemming Further simplify tokens by removing common suffixes like "ing," "ed," or "s," reducing words to their root form.
4. Stop-Word Removal Eliminate commonly used words (e.g., "is," "to," "the") that add little semantic value to the analysis.
5. Word2Vec Embedding Generate word embeddings by creating fixed-size vector representations for each unique token. These vectors capture semantic meaning and relationships between words.
6. Word Index Mapping Map each unique token to an index and create a reverse mapping for efficient token-to-index conversion. These indices can be used for embedding layers or other processing steps.

Step 1: Import Necessary Libraries[¶](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg#Step-1:-Import-Necessary-Libraries)

In [ ]:

Student's answer[(Top)](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg" \l "top)

*# YOUR CODE HERE*

**import** **string**

**import** **torch**

**import** **torch.nn** **as** **nn**

**import** **torch.optim** **as** **optim**

Step 2: Define the Paragraph[¶](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg#Step-2:-Define-the-Paragraph)

You are provided with a paragraph of text about machine learning, store it in a variable named paragraph. Machine learning is fascinating. It allows computers to learn from data. The more data, the better the learning. Deep learning is a subset of machine learning. Neural networks are at the core of deep learning. Artificial intelligence is evolving rapidly. Data science combines domain expertise with programming skills. Big data plays a CRUCIAL role in MODERN analytics. Natural language processing is a key part of AI. Predictive modeling helps in forecasting future trends."""

In [ ]:

paragraph = """

Machine learning is fascinating.

It allows computers to learn from data.

The more data, the better the learning.

Deep learning is a subset of machine learning.

Neural networks are at the core of deep learning.

Artificial intelligence is evolving rapidly.

Data science combines domain expertise with programming skills.

Big data plays a CRUCIAL role in MODERN analytics.

Natural language processing is a key part of AI.

Predictive modeling helps in forecasting future trends."""

Step 3: Tokenization and Lowercasing[¶](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg#Step-3:-Tokenization-and-Lowercasing)

In this task, you will implement a function to tokenize a given paragraph of text and convert the tokens to lowercase. Tokenization involves splitting the input text into individual words (tokens) while removing punctuation and whitespace. Converting tokens to lowercase ensures uniformity and consistency in text processing, which is essential for natural language processing tasks. Steps to Tokenize Text: 1. Initialize an Empty List (tokens) and an Empty String (word): Create an empty list, tokens, to store the resulting tokens. Create an empty string, word, to accumulate characters that form a word. 2. Iterate Through Each Character in the Input Text: Loop through each character (char) in the given text. 3. Check for Word Boundaries: If the character is a whitespace or punctuation (indicating the end of a word): Check if word contains any accumulated characters: If it does, append the word to the tokens list. Reset word to an empty string to start accumulatin1g the next word. Else If the character is not a whitespace or punctuation: Append the character to the word string (accumulate characters for the current word). 4. Add the Last Word to Tokens (if any): After the loop finishes, check if there is a non-empty word: If word contains characters, append it to the tokens list. This step ensures that any word being accumulated at the end of the text is not missed. Return the List of Tokens: Return the tokens list, which now contains all the individual words extracted from the input text. 5. Tokenize the Paragraph: Call the tokenize function, passing the paragraph as input. This function processes the paragraph and returns the list tokens, which contains all the individual words extracted from the text. 6.Convert Tokens to Lowercase: Iterate through each word in the tokens list. Use the .lower() method to convert each word to lowercase. Create a new list, lower\_tokens, to store the resulting lowercase word Function Signature: def tokenize(text): # Your code here return tokens Parameters: text: A string containing the input text to tokenize. Expected Output: tokens: A list of words (tokens) extracted from the input text, excluding punctuation and split by whitespace.

In [ ]:

Student's answer[(Top)](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg" \l "top)

*# YOUR CODE HERE*

**def** tokenize(text):

tokens = []

word = ""

**for** char **in** text:

**if** char **in** string.whitespace **or** char **in** string.punctuation:

**if** word:

tokens.append(word)

word = ""

**else**:

word += char

**if** word:

tokens.append(word)

**return** tokens

tokens = tokenize(paragraph)

lower\_tokens = [word.lower() **for** word **in** tokens]

print("TOKENS: ", tokens)

print("**\n**LOWER TOKENS: ", lower\_tokens)

Test case 1: Check for tokenization[¶](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg#Test-case-1:-Check-for-tokenization)

In [ ]:

Grade cell: cell-dea8ced7b5063ad6Score: 25.0 / 25.0 [(Top)](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg#top)

Step 4: Lemmatization[¶](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg#Step-4:-Lemmatization)

In this task, you will implement a function to perform lemmatization on individual tokens. 1. Define the Lemmatize Function: Create a function named lemmatize that takes a single input, token. Inside the function, define a dictionary called lemmas that maps specific words to their lemmas (base forms).(example: 'computers' must be mapped as 'computer') 2.Retrieve Lemmas Using the Dictionary: Use the lemmas.get(token, token) method: If the input token exists as a key in the lemmas dictionary, return its corresponding value (the lemma). If the token is not in the dictionary, return the original token as the default. 3. Apply Lemmatization to Tokens: Use a list comprehension to process each word in the lower\_tokens list. For each token, call the lemmatize function to replace it with its lemma (if available). 4.Store the Results: Store the output of the list comprehension in a new list called lemmatized\_tokens.Function Signature: def lemmatize(token): # Your code here return lemmas.get(token, token) Parameters: token: A single word that needs to be lemmatized. Expected Output: lemma: The base or root form of the word.

In [ ]:

Student's answer[(Top)](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg" \l "top)

*# YOUR CODE HERE*

**def** lemmatize(token):

lemmas = {

'learning': 'learn',

'computers': 'computer',

'data': 'datum',

'networks': 'network'

}

**return** lemmas.get(token, token)

lemmatized\_tokens = [lemmatize(token) **for** token **in** lower\_tokens]

print("LEMMATIZED TOKENS: ", lemmatized\_tokens)

Test case 2: Check for lemmatization[¶](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg#Test-case-2:-Check-for-lemmatization)

In [ ]:

Grade cell: cell-fecaffd35d0f0438Score: 25.0 / 25.0 [(Top)](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg#top)

Step 5: Stemming[¶](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg#Step-5:-Stemming)

In this task, you will implement a function to perform stemming on individual tokens.

def stem(token): # Your code here return token # Your code here1. Define the Stemming Function: Create a function named stem that takes a single input, token. Inside the function, define a list called suffixes that contains common suffixes to remove ('ing', 'ed', 's'). 2. Check for Suffixes: Iterate through each suffix in the suffixes list. For each suffix: Check if the token ends with that suffix using token.endswith(suffix). If it does, remove the suffix by slicing the token: token[:-len(suffix)]. 3. Handle Tokens Without Suffixes: If the token does not end with any of the specified suffixes, return it unchanged. 4. Apply Stemming to Tokens: Use a list comprehension to process each word in the lemmatized\_tokens list. For each token, call the stem function to remove applicable suffixes. 5. Store the Results: Store the output of the list comprehension in a new list called stemmed\_tokens.

In [ ]:

Student's answer[(Top)](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg" \l "top)

*# YOUR CODE HERE*

**def** stem(token):

suffixes = ['ing', 'ed', 's']

**for** suffix **in** suffixes:

**if** token.endswith(suffix):

**return** token[:-len(suffix)]

**return** token

stemmed\_tokens = [stem(token) **for** token **in** lemmatized\_tokens]

print("STEMMED TOKENS: ", stemmed\_tokens)

Test case 3: Check for stemming[¶](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg#Test-case-3:-Check-for-stemming)

In [ ]:

Grade cell: cell-2b3c0e9cce5e4ba5Score: 25.0 / 25.0 [(Top)](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg#top)

Step 6:Remove Stop Words[¶](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg#Step-6:Remove-Stop-Words)

In this task, you will remove stop words from a list of tokens.

In [ ]:

Student's answer[(Top)](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg" \l "top)

*# YOUR CODE HERE*

stop\_words = {'is', 'to', 'the', 'from', 'and', 'are', 'at', 'of', 'a'}

filtered\_tokens = [token **for** token **in** stemmed\_tokens **if** token **not** **in** stop\_words]

print("FILTERED TOKENS: ", filtered\_tokens)

STEP 7: Word2Vec Embedding[¶](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg#STEP-7:--Word2Vec-Embedding)

In this task, you will implement a function to create word embeddings using Word2Vec-like vector representations. Word embeddinygs are dense numerical representations of words in a fixed-dimensional vector space.

def create\_word\_vectors(tokens, vector\_size=50): # Your code here return word\_vectors

In [ ]:

Student's answer[(Top)](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg" \l "top)

*# YOUR CODE HERE*

**def** create\_word\_vectors(tokens, vector\_size=50):

vocab = set(tokens)

word\_vectors = {word: torch.rand(vector\_size) **for** word **in** vocab}

**return** word\_vectors

Step 8. Assign Word Indices[¶](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg#Step-8.-Assign-Word-Indices)

In this task, you will assign unique indices to each word in the filtered tokens list. This step involves creating two dictionaries: one to map words to their respective indices and another to map indices back to their corresponding words. This mapping is a crucial step in preparing textual data for machine learning models

In [ ]:

Student's answer[(Top)](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg" \l "top)

*# YOUR CODE HERE*

word\_vectors = create\_word\_vectors(filtered\_tokens)

word\_to\_index = {word: idx **for** idx, word **in** enumerate(filtered\_tokens)}

print("WORD TO INDEX: ", word\_to\_index)

Test case 4: Check for word2vec embedding[¶](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg#Test-case-4:-Check-for-word2vec-embedding)

In [ ]:

Grade cell: cell-8440f3e352a5caf8Score: 25.0 / 25.0 [(Top)](http://10.11.51.204:8000/user/23bd1a054t/files/ps2/Tokenization/feedback/2025-01-27%2005%3A24%3A02.519276%20UTC/Tokenizataion-1.html?_xsrf=MnwxOjB8MTA6MTczNzk2NjQ3Nnw1Ol94c3JmfDEzMjpPV000TmpVeE1EVmtZalEyTkdFMk9UbGtNakExWlRNMk5XRXlZekl3T0RJNk5EUXpOamxrTmpBeU1qRmlZMkpsTjJSbU1EQTBPVFk1TlRsa05UbGlOakUwTkdGa01qUXpNbUZsTnpsaFkyUTRNbVF3WXpObFptTTROekF3WWpRd05nPT18MGM0MTJmZmJkNTE1M2Y4MDNjYjM2MjdhZjJjODdmZTVlYWUwNDFmNzdlY2QxYWQyYjRmMTI5MmNhMjE2OTY3Mg#top)

In [ ]:

In [ ]: