Why we use case normalization when tokenizing?

- We use case normalization to reduce the total number of tokens by not having upper and lower case duplicates. This helps in reducing the dimensionality of our data and simplifying our models. For example, without case normalization, "Coffee" and "coffee" would be treated as different tokens even though they are the same word.

True statement about the provided tokens.

- The list of tokens were created by splitting on the white space between the words in the sentence. This is a simple way to tokenize a sentence - by treating each space-separated segment as a separate token.

How would you determine which words to use in a customized list of stop words?

- The idea is to identify words that are so common in the corpus that they do not provide much meaningful information for analysis. By ranking all the words by their frequency and identifying stop words as those above a certain percentage, we can eliminate these common words and focus on the words that are more likely to be meaningful.

What is the process of statistical trimming?

- Statistical trimming is the process of looking at the distribution of word counts and then "trimming" off both the most and least common words. The most common words are often stop words, and the least common words may be typos or very rare words that won't provide much insight.

What is the difference between stemming and lemmatization?

- Stemming is the process of reducing a word to its base or root form, often by removing suffixes. Lemmatization, on the other hand, reduces words to their base form according to the dictionary definition of the word. This is more complex than stemming as it involves understanding the context in which a word is used.

Are the provided word changes a result of stemming or lemmatization or neither?

- The provided word changes are a result of lemmatization. Lemmatization would reduce "falling" to "fall" and "went" to "go" based on the grammatical rules of English. Stemming could also reduce "falling" to "fall", but it would not change "went" to "go".

How many reviews are there for "Summer Moon Coffee Bar"?

- This answer would require a look into the data. The code to find this would be something like len(df[df['coffee_shop_name'] == 'Summer Moon Coffee Bar']) or `len(df[df['coffee_shop_name'].str.contains('Summer Moon Coffee Bar', case=False, na=False)])` if the string was not exact .

What do the following lines of code do? The text variable is a string of characters.

- The first line uses a regular expression to keep only the lower case letters, upper case letters, spaces, and numbers. The second line converts all characters to lowercase and splits the string into individual words based on spaces.

How would we apply a tokenization function called tokenize to a column of a DataFrame?

- The apply() function is used to apply a function along an axis of the DataFrame. So, shops['review_text'].apply(tokenize) applies the tokenize function to every row in the 'review_text' column of the DataFrame.

What does the following code do if provided a list of tokens (tokens)?

- The provided code checks each token to see if it is not a stop word and not punctuation. If both conditions are true, it lowercases the token and appends it to a list. This way, it filters out stop words and punctuation, and ensures all tokens are in lower case.

What does the following code do? Assume you have a document defines as doc which is a collection of tokens.

- This code loops over each token in the document and appends the lemma of the token to the list 'lemmas'. Lemmatization reduces words to their base or root form (lemma), so this code effectively lemmatizes the document.