

Start coding or generate with AI.

Task

Perform text preprocessing on a user-provided paragraph by installing and importing necessary libraries (NLTK, spaCy), loading the text, counting the total number of words, converting the text to lowercase, and finally summarizing the performed preprocessing steps, including the word count and the lowercased text.

Install and Import Libraries

Subtask:

Install the NLTK and spaCy libraries. Then, import these libraries along with any other necessary modules for text processing.

Reasoning: First, I need to install the `nltk` library as per the instructions. This requires a separate code block for the `!pip install` command.

```
!pip install nltk

Requirement already satisfied: nltk in /usr/local/lib/python3.12/dist-packages (3.9.1)
Requirement already satisfied: click in /usr/local/lib/python3.12/dist-packages (from nltk) (8.3.1)
Requirement already satisfied: joblib in /usr/local/lib/python3.12/dist-packages (from nltk) (1.5.3)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.12/dist-packages (from nltk) (2025.11.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from nltk) (4.67.1)
```

Reasoning: Next, I need to install the `spaCy` library as per the instructions. This requires a separate code block for the `!pip install` command.

```
import sys
!pip install spacy

Requirement already satisfied: spacy in /usr/local/lib/python3.12/dist-packages (3.8.11)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.0.15)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.13)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.0.12)
Requirement already satisfied: thinc<8.4.0,>=8.3.4 in /usr/local/lib/python3.12/dist-packages (from spacy) (8.3.10)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.1.3)
Requirement already satisfied: srslx<3.0.0,>=2.4.3 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.5.2)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.10)
Requirement already satisfied: weasel<0.5.0,>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (0.4.3)
Requirement already satisfied: typer-slim<1.0.0,>=0.3.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (0.20.0)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (4.67.1)
Requirement already satisfied: numpy>=1.19.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.2)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.32.4)
Requirement already satisfied: pydantic!=1.8.=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.12/dist-packages (from spacy)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.1.6)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from spacy) (75.2.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (25.0)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8.=1.8.1)
Requirement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8.=1.8.1)
Requirement already satisfied: typing-extensions>=4.14.1 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8.=1.8)
Requirement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8.=1.8)
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spacy)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spacy)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spacy)
Requirement already satisfied: blis<1.4.0,>=1.3.0 in /usr/local/lib/python3.12/dist-packages (from thinc<8.4.0,>=8.3.4->spacy)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.12/dist-packages (from thinc<8.4.0,>=8.3.4)
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.12/dist-packages (from typer-slim<1.0.0,>=0.3.0->spacy)
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.12/dist-packages (from weasel<0.5.0,>=0.4.2)
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.12/dist-packages (from weasel<0.5.0,>=0.4.2)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->spacy) (3.0.3)
Requirement already satisfied: wrapt in /usr/local/lib/python3.12/dist-packages (from smart-open<8.0.0,>=5.2.1->weasel<0.5.0)
```

Reasoning: Now that both NLTK and spaCy are installed, I will import them as instructed. Additionally, I need to download the 'punkt' tokenizer data for NLTK, which is a necessary step for many NLTK text processing functionalities.

```
import nltk
import spacy
```

```
nltk.download('punkt')

print("NLTK and spaCy imported, and NLTK 'punkt' data downloaded.")

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]  Unzipping tokenizers/punkt.zip.
NLTK and spaCy imported, and NLTK 'punkt' data downloaded.
```

Reasoning: The previous steps successfully installed and imported the necessary libraries (NLTK and spaCy) and downloaded the 'punkt' tokenizer data. The next logical step, as per the task description, is to load the user-provided text for preprocessing.

```
user_paragraph = "This is an example paragraph that will be used for text preprocessing. It contains several sentences and n
print("User paragraph loaded:")
print(user_paragraph)

User paragraph loaded:
This is an example paragraph that will be used for text preprocessing. It contains several sentences and needs to be cleaned
```

▼ Count Words

Subtask:

Count the total number of words in the loaded text using appropriate Python string methods or NLTK functionalities.

Reasoning: As per the instructions, I will use `nltk.word_tokenize()` to break down the `user_paragraph` into individual words and then use `len()` to get the total word count. Finally, I will print the `word_count`.

```
words = nltk.word_tokenize(user_paragraph)
word_count = len(words)

print(f"Total word count: {word_count}")

Total word count: 25
```

Reasoning: The previous code failed because NLTK's `punkt_tab` resource was not found. To fix this, I need to download the `punkt_tab` tokenizer data, which is explicitly suggested by the error message and is a necessary prerequisite for `nltk.word_tokenize()` to function correctly in this context.

```
import nltk
nltk.download('punkt_tab')
print("NLTK 'punkt_tab' data downloaded, which is required for tokenization.")

NLTK 'punkt_tab' data downloaded, which is required for tokenization.
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]  Unzipping tokenizers/punkt_tab.zip.
```

Reasoning: The previous error indicated that the `punkt_tab` resource was missing, which has now been downloaded. I will re-run the word counting code, as the prerequisite for `nltk.word_tokenize()` is now met.

```
words = nltk.word_tokenize(user_paragraph)
word_count = len(words)

print(f"Total word count: {word_count}")

Total word count: 25
```

▼ Convert to Lowercase

Subtask:

Convert the entire text to lowercase using Python's built-in string methods.

Reasoning: As per the instructions, I need to convert the `user_paragraph` to lowercase using the `.lower()` method, store it in `lower_case_paragraph`, and then print the result.

```
lower_case_paragraph = user_paragraph.lower()

print("Lowercased paragraph:")
print(lower_case_paragraph)
```

Lowercased paragraph:
this is an example paragraph that will be used for text preprocessing. it contains several sentences and needs to be cleaned