program 2. Using Python libraries, download Wikipedia's page on open source and tokenize the text, remove the stop words. What percentage of the page is stop words?

Objective:

The program aims to analyze the proportion of stop words in a Wikipedia article by:

Fetching the Wikipedia summary of "Open Source" using the Wikipedia API.

Tokenizing the text into individual words using the spaCy NLP library.

Identifying and removing stop words (commonly used words like "the," "is," and "and" that do not carry significant meaning).

Calculating the percentage of stop words in the text.

Methodology:

The Wikipedia content is retrieved programmatically.

spaCy is used to tokenize the text and filter out stop words.

The total number of words and stop words are counted.

Finally, the percentage of stop words in the text is computed.

1. Download Wikipedia Page Content

```
import requests  # To fetch Wikipedia content

url = "https://en.wikipedia.org/api/rest_v1/page/summary/Open_source"
response = requests.get(url)  # Sending HTTP GET request
data = response.json()  # Converting response to JSON format
text = data["extract"]  # Extracting the main content
```

We use the Wikipedia API to fetch the summary of the "Open Source" page.

The response is in JSON format, and we extract the main text from the "extract" key.

2. Load spaCy's English Model

```
nlp = spacy.load("en_core_web_sm")
```

We load the pre-trained small English model in spaCy, which includes tokenization and stop-word detection.

3. Process the Text

```
doc = nlp(text)
```

The Wikipedia text is processed using spaCy, and it returns a Doc object containing structured NLP tokens.

4. Tokenization and Stop-word Removal

```
tokens = [token.text for token in doc]  # Tokenizing the text
stop_words = [token.text for token in doc if token.is_stop]  # Extracting stop words
```

doc contains individual tokens (words and punctuation). We extract all tokens and separately extract only stop words using token.is_stop.

5. Calculate Percentage of Stop Words

```
total_tokens = len(tokens)
stop_word_count = len(stop_words)
percentage_stop_words = (stop_word_count / total_tokens) * 100
```

We count the total words and stop words. The percentage of stop words is calculated as: $\left(\frac{\text{stop word count}}{\text{total token count}}\right) \times 100$

6. Display Results

```
print(f"Total words: {total_tokens}")
print(f"Stop words: {stop_word_count}")
print(f"Percentage of stop words: {percentage_stop_words:.2f}%")
```

```
Total words: 118
Stop words: 40
Percentage of stop words: 33.90%
```

The total words, stop words, and percentage of stop words are displayed.

```
import requests  # To fetch Wikipedia content
import spacy  # For NLP processing

# Step 1: Download Wikipedia page content
url = "https://en.wikipedia.org/api/rest_v1/page/summary/Open_source"
```

```
response = requests.get(url)  # Sending HTTP GET request
data = response.json()  # Converting response to JSON format
text = data["extract"]  # Extracting the main content

# Step 2: Load spaCy's English model
nlp = spacy.load("en_core_web_sm")

# Step 3: Process text using spaCy
doc = nlp(text)

# Step 4: Tokenization and Stop-word Removal
tokens = [token.text for token in doc]  # Tokenizing the text
stop_words = [token.text for token in doc if token.is_stop]  # Extracting stop words

# Step 5: Calculate Percentage of Stop Words
total_tokens = len(tokens)
stop_word_count = len(stop_words)
percentage_stop_words = (stop_word_count / total_tokens) * 100

# Display results
print(f"Total words: {total_tokens}")
print(f"Stop words: {stop_word_count}")
print(f"Percentage of stop words: {percentage_stop_words:.2f}%")
```

```
Total words: 118
Stop words: 40
Percentage of stop words: 33.90%
```