Implementation with NLTK

NLTK provides robust support for stopword removal across 16 different languages. The implementation involves tokenization followed by filtering:

Setup: Import NLTK modules and download required resources like stopwords and tokenizer data.

Text preprocessing: Convert the sample sentence to lowercase and tokenize it into words.

Stopword removal: Load English stopwords and filter them out from the token list.

Output: Print both the original and cleaned tokens for comp

In [2]:
```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

nltk.download('punkt')
nltk.download('stopwords')
nltk.download('punkt_tab') # Added to resolve LookupError: punkt_tab

text = "This is a sample sentence showing stopword removal."
words = word_tokenize(text.lower())

stop_words = set(stopwords.words('english'))
filtered_words = [word for word in words if word.casefold() not in stop_words]

print("Original Words:", words)
print("Filtered Words:", filtered_words)
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
Original Words: ['this', 'is', 'a', 'sample', 'sentence', 'showing', 'stopwor
d', 'removal', '.']
Filtered Words: ['sample', 'sentence', 'showing', 'stopword', 'removal', '.']
```

Implementation using SpaCy

SpaCy offers a more sophisticated approach with built-in linguistic analysis:

Imports spaCy: Used for natural language processing.

Load model: Loads the English NLP model with tokenization and stopword detection.

Process text: Converts the sentence into a Doc object with linguistic features.

Remove stopwords: Filters out common words using token.is_stop.

Print output: Displays non-stopword tokens like ['researchers', 'developing', 'advanced', 'algorithms'].

In [3]:
```
import spacy

nlp = spacy.load("en_core_web_sm")
doc = nlp("This is a sample sentence showing stopword removal.")

filtered_tokens = [token.text for token in doc if not token.is_stop]

print("Filtered Tokens:", filtered_tokens)
```

```
Filtered Tokens: ['sample', 'sentence', 'showing', 'stopword', 'removal', '.']
```

Removing stop words with Genism

Import function: Brings in remove_stopwords from Gensim.

Define text: A sample sentence is used.

Apply stopword removal: Removes common words like “the,” “a”.

Print output: Shows original and filtered text.

In [5]:
```
!pip install gensim
from gensim.parsing.preprocessing import remove_stopwords

text = "This is a sample sentence showing stopword removal."
filtered_text = remove_stopwords(text)

print("Original Text:", text)
print("Filtered Text:", filtered_text)
```

```
Collecting gensim
  Downloading gensim-4.4.0-cp312-cp312-manylinux_2_24_x86_64.manylinux_2_28_x8
6_64.whl.metadata (8.4 kB)
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.12/dist-
packages (from gensim) (2.0.2)
Requirement already satisfied: scipy>=1.7.0 in /usr/local/lib/python3.12/dist-p
ackages (from gensim) (1.16.3)
Requirement already satisfied: smart_open>=1.8.1 in /usr/local/lib/python3.12/d
ist-packages (from gensim) (7.5.0)
Requirement already satisfied: wrapt in /usr/local/lib/python3.12/dist-packages
(from smart_open>=1.8.1->gensim) (2.0.1)
Downloading gensim-4.4.0-cp312-cp312-manylinux_2_24_x86_64.manylinux_2_28_x86_6
4.whl (27.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 27.9/27.9 MB 66.2 MB/s eta 0:00:00
Installing collected packages: gensim
Successfully installed gensim-4.4.0
Original Text: This is a sample sentence showing stopword removal.
Filtered Text: This sample sentence showing stopword removal.
```

Implementation with Scikit Learn

Imports necessary modules from sklearn and nltk for tokenization and stopword removal.

Defines a sample sentence

Tokenizes the sentence into individual words using NLTK's word_tokenize.

Filters out common English stopwords from the token list.

Prints both the original and stopword-removed versions of the text.

In [8]:
```python
from sklearn.feature_extraction.text import CountVectorizer
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

new_text = "The quick brown fox jumps over the lazy dog."

new_word = word_tokenize(new_text)

new_filtered_word = [word for word in new_word if word.casefold() not in stopw

new_clean_text = ' '.join(new_filtered_word)

print("Original Words:", new_text)
print("Filtered Words:", new_clean_text)
```

```
Original Words: The quick brown fox jumps over the lazy dog.
Filtered Words: quick brown fox jumps lazy dog .
```