PREPROCESSING TEXT DATASET

NAME: TAMANNA VAIKKATH

```
!pip install nltk --quiet

import nltk

nltk.download('punkt', force=True)
nltk.download('stopwords', force=True)
nltk.download('wordnet', force=True)
nltk.download('omw-1.4', force=True)
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
True
```

```
!pip install -U spacy --quiet
!python -m spacy download en_core_web_sm

import spacy
from nltk.stem import PorterStemmer


nlp = spacy.load("en_core_web_sm")
stemmer = PorterStemmer()


text_data = [
    "Artificial Intelligence is transforming the future.",
    "Natural Language Processing allows machines to understand human language.",
    "Search algorithms help in solving complex problems efficiently."
]

for i, sentence in enumerate(text_data):
    print(f"\n  Sentence {i+1}: {sentence}")
    doc = nlp(sentence)

    # Tokenization
    tokens = [token.text for token in doc]
    print(f"  Tokens: {tokens}")

    # Stopword Removal
    filtered = [token for token in doc if not token.is_stop and token.is_alpha]
    filtered_text = [token.text for token in filtered]
    print(f"  After Stopword Removal: {filtered_text}")

    # Lemmatization
    lemmatized = [token.lemma_ for token in filtered]
    print(f"  Lemmatized Tokens: {lemmatized}")

    # Stemming (optional with NLTK)
    stemmed = [stemmer.stem(token.text) for token in filtered]
    print(f"  Stemmed Tokens: {stemmed}")
```

```
Collecting en-core-web-sm==3.8.0
  Downloading https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-3.8.0/en_core_web_sm-3.8.0-py3-none-any.whl (12
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.8/12.8 MB 50.4 MB/s eta 0:00:00
✓ Download and installation successful
You can now load the package via spacy.load('en_core_web_sm')
⚠ Restart to reload dependencies
If you are in a Jupyter or Colab notebook, you may need to restart Python in
order to load all the package's dependencies. You can do this by selecting the
'Restart kernel' or 'Restart runtime' option.

  Sentence 1: Artificial Intelligence is transforming the future.
  Tokens: ['Artificial', 'Intelligence', 'is', 'transforming', 'the', 'future', '.']
  After Stopword Removal: ['Artificial', 'Intelligence', 'transforming', 'future']
```

```
Lemmatized Tokens: ['Artificial', 'Intelligence', 'transform', 'future']
Stemmed Tokens: ['artifici', 'intellig', 'transform', 'futur']

Sentence 2: Natural Language Processing allows machines to understand human language.
Tokens: ['Natural', 'Language', 'Processing', 'allows', 'machines', 'to', 'understand', 'human', 'language', '.']
After Stopword Removal: ['Natural', 'Language', 'Processing', 'allows', 'machines', 'understand', 'human', 'language']
Lemmatized Tokens: ['Natural', 'Language', 'processing', 'allow', 'machine', 'understand', 'human', 'language']
Stemmed Tokens: ['natur', 'languag', 'process', 'allow', 'machin', 'understand', 'human', 'languag']

Sentence 3: Search algorithms help in solving complex problems efficiently.
Tokens: ['Search', 'algorithms', 'help', 'in', 'solving', 'complex', 'problems', 'efficiently', '.']
After Stopword Removal: ['Search', 'algorithms', 'help', 'solving', 'complex', 'problems', 'efficiently']
Lemmatized Tokens: ['search', 'algorithm', 'help', 'solve', 'complex', 'problem', 'efficiently']
Stemmed Tokens: ['search', 'algorithm', 'help', 'solv', 'complex', 'problem', 'effici']
```