```
# ------------------------------------------------------------
# Practical Assignment 4 - Text Preprocessing
# ------------------------------------------------------------
# Objective: Perform tokenization, stemming, and stop-word removal
# along with other cleaning steps on text data.
```

```
# Step 1: Import Libraries
!pip install emoji

import re
import string
import emoji
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer, WordNetLemmatizer

import nltk
nltk.download('punkt')
nltk.download('punkt_tab')

nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
```

```
Requirement already satisfied: emoji in /usr/local/lib/python3.12/dist-packages (2.15.0)
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
[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 wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
True
```

```
# ------------------------------------------------------------
# Step 2: Sample Text Dataset (can be replaced with any dataset)
texts = [
    "Wow!!! This movie was AMAZING 😍😍 Visit https://imdb.com for more info.",
    "I didn't like the film... It was boring and too long!!! #wasteoftime",
    "Machine Learning is improving automation in industries. 100% effective!",
    "The actors were great, but the story was average. 🙄",
]

print("Original Texts:\n")
for t in texts:
    print("-", t)
```

```
Original Texts:

- Wow!!! This movie was AMAZING 😍😍 Visit https://imdb.com for more info.
- I didn't like the film... It was boring and too long!!! #wasteoftime
- Machine Learning is improving automation in industries. 100% effective!
- The actors were great, but the story was average. 🙄
```

```
# ------------------------------------------------------------
# Step 3: Define Preprocessing Function
def preprocess_text(text):
    print("\n==============================")
    print("Original Text:", text)

    # 1. Text Cleaning - remove URLs, emojis, hashtags, etc.
    text = re.sub(r"http\S+|www\S+", "", text)  # Remove URLs
    text = emoji.replace_emoji(text, replace='')  # Remove emojis
    text = re.sub(r"#\w+", "", text)  # Remove hashtags

    # 2. Lowercasing
    text = text.lower()

    # 3. Remove punctuation and numbers
    text = text.translate(str.maketrans('', '', string.punctuation))
    text = re.sub(r'\d+', '', text)

    # 4. Tokenization
    tokens = word_tokenize(text)
    # 5. Stop-word Removal
    stop_words = set(stopwords.words('english'))
    filtered_tokens = [w for w in tokens if w not in stop_words]

    # 6. Stemming
    ps = PorterStemmer()
    stemmed_tokens = [ps.stem(w) for w in filtered_tokens]

    # 7. Lemmatization
    lemmatizer = WordNetLemmatizer()
    lemmatized_tokens = [lemmatizer.lemmatize(w) for w in filtered_tokens]

    # Display intermediate outputs
    print("Cleaned Text:", text)
    print("Tokens:", tokens)
    print("Filtered (No Stop-words):", filtered_tokens)
    print("Stemmed Tokens:", stemmed_tokens)
    print("Lemmatized Tokens:", lemmatized_tokens)

    # Return processed text
    return " ".join(lemmatized_tokens)
```

```
# ------------------------------------------------------------
# Step 4: Apply Preprocessing
cleaned_texts = [preprocess_text(t) for t in texts]
```

```
==============================
Original Text: Wow!!! This movie was AMAZING 😍😍 Visit https://imdb.com for more info.
Cleaned Text: wow this movie was amazing  visit  for more info
Tokens: ['wow', 'this', 'movie', 'was', 'amazing', 'visit', 'for', 'more', 'info']
Filtered (No Stop-words): ['wow', 'movie', 'amazing', 'visit', 'info']
Stemmed Tokens: ['wow', 'movi', 'amaz', 'visit', 'info']
Lemmatized Tokens: ['wow', 'movie', 'amazing', 'visit', 'info']

==============================
Original Text: I didn't like the film... It was boring and too long!!! #wasteoftime
Cleaned Text: i didnt like the film it was boring and too long
Tokens: ['i', 'didnt', 'like', 'the', 'film', 'it', 'was', 'boring', 'and', 'too', 'long']
Filtered (No Stop-words): ['didnt', 'like', 'film', 'boring', 'long']
Stemmed Tokens: ['didnt', 'like', 'film', 'bore', 'long']
Lemmatized Tokens: ['didnt', 'like', 'film', 'boring', 'long']

==============================
Original Text: Machine Learning is improving automation in industries. 100% effective!
Cleaned Text: machine learning is improving automation in industries  effective
Tokens: ['machine', 'learning', 'is', 'improving', 'automation', 'in', 'industries', 'effective']
Filtered (No Stop-words): ['machine', 'learning', 'improving', 'automation', 'industries', 'effective']
Stemmed Tokens: ['machin', 'learn', 'improv', 'autom', 'industri', 'effect']
Lemmatized Tokens: ['machine', 'learning', 'improving', 'automation', 'industry', 'effective']

==============================
Original Text: The actors were great, but the story was average. 🙄
Cleaned Text: the actors were great but the story was average
Tokens: ['the', 'actors', 'were', 'great', 'but', 'the', 'story', 'was', 'average']
Filtered (No Stop-words): ['actors', 'great', 'story', 'average']
Stemmed Tokens: ['actor', 'great', 'stori', 'averag']
Lemmatized Tokens: ['actor', 'great', 'story', 'average']
```

```
# ------------------------------------------------------------
# Step 4: Apply Preprocessing
cleaned_texts = [preprocess_text(t) for t in texts]
```

```
==============================
Original Text: Wow!!! This movie was AMAZING 😍😍 Visit https://imdb.com for more info.
Cleaned Text: wow this movie was amazing  visit  for more info
Tokens: ['wow', 'this', 'movie', 'was', 'amazing', 'visit', 'for', 'more', 'info']
Filtered (No Stop-words): ['wow', 'movie', 'amazing', 'visit', 'info']
Stemmed Tokens: ['wow', 'movi', 'amaz', 'visit', 'info']
Lemmatized Tokens: ['wow', 'movie', 'amazing', 'visit', 'info']

==============================
Original Text: I didn't like the film... It was boring and too long!!! #wasteoftime
Cleaned Text: i didnt like the film it was boring and too long
Tokens: ['i', 'didnt', 'like', 'the', 'film', 'it', 'was', 'boring', 'and', 'too', 'long']
Filtered (No Stop-words): ['didnt', 'like', 'film', 'boring', 'long']
Stemmed Tokens: ['didnt', 'like', 'film', 'bore', 'long']
Lemmatized Tokens: ['didnt', 'like', 'film', 'boring', 'long']

==============================
Original Text: Machine Learning is improving automation in industries. 100% effective!
Cleaned Text: machine learning is improving automation in industries  effective
```

```
Tokens: ['machine', 'learning', 'is', 'improving', 'automation', 'in', 'industries', 'effective']
Filtered (No Stop-words): ['machine', 'learning', 'improving', 'automation', 'industries', 'effective']
Stemmed Tokens: ['machin', 'learn', 'improv', 'autom', 'industri', 'effect']
Lemmatized Tokens: ['machine', 'learning', 'improving', 'automation', 'industry', 'effective']

==============================
Original Text: The actors were great, but the story was average. 🙄
Cleaned Text: the actors were great but the story was average
Tokens: ['the', 'actors', 'were', 'great', 'but', 'the', 'story', 'was', 'average']
Filtered (No Stop-words): ['actors', 'great', 'story', 'average']
Stemmed Tokens: ['actor', 'great', 'stori', 'averag']
Lemmatized Tokens: ['actor', 'great', 'story', 'average']
```

```
# ------------------------------------------------------------
# Step 5: Results
print("\n\nFinal Preprocessed Texts:\n")
for i, ct in enumerate(cleaned_texts, 1):
    print(f"{i}. {ct}")
```

```


Final Preprocessed Texts:

1. wow movie amazing visit info
2. didnt like film boring long
3. machine learning improving automation industry effective
4. actor great story average
```

```
# ------------------------------------------------------------
# Step 6: Analysis and Conclusion
print("\nAnalysis:")
print("1. Text cleaning removed unwanted URLs, emojis, and punctuation effectively.")
print("2. Stop-word removal and stemming reduced text size while keeping meaning intact.")
print("3. Lemmatization provided proper word forms, making text more uniform for modeling.")

print("\nConclusion:")
print("Text preprocessing effectively transforms raw, unstructured text into clean, analyzable form, improving model perfor
```

```

Analysis:
1. Text cleaning removed unwanted URLs, emojis, and punctuation effectively.
2. Stop-word removal and stemming reduced text size while keeping meaning intact.
3. Lemmatization provided proper word forms, making text more uniform for modeling.

Conclusion:
Text preprocessing effectively transforms raw, unstructured text into clean, analyzable form, improving model performance ir
```