```
# Text Preprocessing Pipeline using NLTK
import pandas as pd
import re
import nltk

from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
```

```
# Download required NLTK resources
# ------------------------------
nltk.download('punkt')
nltk.download('punkt_tab')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.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
```

```
# Load Dataset (first 1000 rows)
# ------------------------------
df = pd.read_csv('arxiv_data.csv', engine='python', nrows=1000)
print(df.head())
```

```
                                              titles  \
0  Survey on Semantic Stereo Matching / Semantic ...
1  FUTURE-AI: Guiding Principles and Consensus Re...
2  Enforcing Mutual Consistency of Hard Regions f...
3  Parameter Decoupling Strategy for Semi-supervi...
4  Background-Foreground Segmentation for Interio...

                                           summaries  \
0  Stereo matching is one of the widely used tech...
1  The recent advancements in artificial intellig...
2  In this paper, we proposed a novel mutual cons...
3  Consistency training has proven to be an advan...
4  To ensure safety in automated driving, the cor...

                       terms
0          ['cs.CV', 'cs.LG']
1  ['cs.CV', 'cs.AI', 'cs.LG']
2          ['cs.CV', 'cs.AI']
3                  ['cs.CV']
4          ['cs.CV', 'cs.LG']
```

```
# Custom text cleaning function using regex
# ------------------------------
def preprocess_text(text):
    if pd.isna(text):
        return ""

    text = re.sub(r'http\S+|www\S+', '', text)        # Remove URLs
    text = re.sub(r'<.*?>', '', text)                 # Remove HTML tags
    text = re.sub(r'@\w+', '', text)                  # Remove mentions
    text = re.sub(r'#\w+', '', text)                  # Remove hashtags
    text = text.lower()                               # Convert to lowercase

    # Remove emojis
    emoji_pattern = re.compile(
        "["
        "\U0001F600-\U0001F64F"
        "\U0001F300-\U0001F5FF"
        "\U0001F680-\U0001F6FF"
        "\U0001F1E0-\U0001F1FF"
        "]+", flags=re.UNICODE)
    text = emoji_pattern.sub(r'', text)

    text = re.sub(r'[^a-zA-Z0-9\s]', '', text)        # Remove special characters
    text = re.sub(r'\s+', ' ', text).strip()          # Remove extra spaces
```

```
    return text

# Apply cleaning
df['processed_summaries'] = df['summaries'].apply(preprocess_text)
```

```
# Tokenization
# ------------------------------
df['tokenized_summaries'] = df['processed_summaries'].apply(word_tokenize)
```

```
# Stopword Removal
# ------------------------------
stop_words = set(stopwords.words('english'))

def remove_stopwords(tokens):
    return [word for word in tokens if word not in stop_words]

df['filtered_summaries'] = df['tokenized_summaries'].apply(remove_stopwords)
```

```
# Lemmatization
# ------------------------------
lemmatizer = WordNetLemmatizer()

def lemmatize_tokens(tokens):
    return [lemmatizer.lemmatize(word) for word in tokens]

df['lemmatized_summaries'] = df['filtered_summaries'].apply(lemmatize_tokens)
```

```
# Rejoin tokens into sentence
# ------------------------------
df['clean_summaries'] = df['lemmatized_summaries'].apply(lambda x: ' '.join(x))
```

```
# Unified NLTK Preprocessing Pipeline
# ------------------------------
def nltk_preprocessing_pipeline(text):
    text = preprocess_text(text)
    tokens = word_tokenize(text)
    tokens = [w for w in tokens if w not in stop_words]
    tokens = [lemmatizer.lemmatize(w) for w in tokens]
    return ' '.join(tokens)
```

```
# Apply unified pipeline
df['clean_summaries_pipeline'] = df['summaries'].apply(nltk_preprocessing_pipeline)
```

```
# Display final output
# ------------------------------
print(df[['summaries', 'clean_summaries_pipeline']].head())
```

```
                                        summaries  \
0  Stereo matching is one of the widely used tech...
1  The recent advancements in artificial intellig...
2  In this paper, we proposed a novel mutual cons...
3  Consistency training has proven to be an advan...
4  To ensure safety in automated driving, the cor...

                          clean_summaries_pipeline
0  stereo matching one widely used technique infe...
1  recent advancement artificial intelligence ai ...
2  paper proposed novel mutual consistency networ...
3  consistency training proven advanced semisuper...
4  ensure safety automated driving correct percep...
```