In [1]:
```
%pip install nltk
```

```
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: nltk in c:\users\pawar\appdata\local\programs\python\python31
3\lib\site-packages (3.9.1)
Requirement already satisfied: click in c:\users\pawar\appdata\local\programs\python\python3
13\lib\site-packages (from nltk) (8.1.8)
Requirement already satisfied: joblib in c:\users\pawar\appdata\local\programs\python\python
313\lib\site-packages (from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in c:\users\pawar\appdata\local\programs\pyth
on\python313\lib\site-packages (from nltk) (2024.11.6)
Requirement already satisfied: tqdm in c:\users\pawar\appdata\local\programs\python\python31
3\lib\site-packages (from nltk) (4.67.1)
Requirement already satisfied: colorama in c:\users\pawar\appdata\local\programs\python\pyth
on313\lib\site-packages (from click->nltk) (0.4.6)
[notice] A new release of pip is available: 24.3.1 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [5]:
```
import nltk
from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer, WordNetLemmatizer
import re

# # Step 1: Download required NLTK packages (handles missing resources)
# nltk.download('punkt_tab')
# nltk.download('stopwords')
# nltk.download('wordnet')
# nltk.download('averaged_perceptron_tagger_eng')

# Step 2: Initialize text
text = "Tokenization is the first step in text analytics. The process of breaking down a te

# Step 3: Perform Tokenization
print("\n--- Tokenization ---")
tokenized_sentences = sent_tokenize(text)  # Sentence Tokenization
tokenized_words = word_tokenize(text)  # Word Tokenization

print("Sentences:", tokenized_sentences)
print("Words:", tokenized_words)

# Step 4: Removing Punctuation & Stop Words
stop_words = set(stopwords.words("english"))

# Remove punctuation and lowercase the text
clean_text = re.sub(r'[^\w\s]', '', text.lower())

# Tokenize and remove stopwords
filtered_words = [word for word in word_tokenize(clean_text) if word not in stop_words]

print("\n--- Stopword Removal ---")
print("Filtered Words:", filtered_words)

# Step 5: Perform Stemming
ps = PorterStemmer()
sample_words = ["wait", "waiting", "waited", "waits"]

print("\n--- Stemming ---")
print([ps.stem(word) for word in sample_words])

# Step 6: Perform Lemmatization
lemmatizer = WordNetLemmatizer()
lem_words = ["studies", "studying", "cries", "cry"]

print("\n--- Lemmatization ---")
```

```
print([lemmatizer.lemmatize(word) for word in lem_words])

# Step 7: Apply POS Tagging
data = "The pink sweater fit her perfectly"
words = word_tokenize(data)

print("\n--- POS Tagging ---")
print(nltk.pos_tag(words))
```

```
--- Tokenization ---
Sentences: ['Tokenization is the first step in text analytics.', 'The process of breaking do
wn a text paragraph into smaller chunks such as words or sentences is called Tokenization.']
Words: ['Tokenization', 'is', 'the', 'first', 'step', 'in', 'text', 'analytics', '.', 'The',
'process', 'of', 'breaking', 'down', 'a', 'text', 'paragraph', 'into', 'smaller', 'chunks',
'such', 'as', 'words', 'or', 'sentences', 'is', 'called', 'Tokenization', '.']

--- Stopword Removal ---
Filtered Words: ['tokenization', 'first', 'step', 'text', 'analytics', 'process', 'breakin
g', 'text', 'paragraph', 'smaller', 'chunks', 'words', 'sentences', 'called', 'tokenizatio
n']

--- Stemming ---
['wait', 'wait', 'wait', 'wait']

--- Lemmatization ---
['study', 'studying', 'cry', 'cry']

--- POS Tagging ---
[('The', 'DT'), ('pink', 'NN'), ('sweater', 'NN'), ('fit', 'VBP'), ('her', 'PRP$'), ('perfec
tly', 'RB')]
```