

NAME: ARJUN T ANIL

REG NO: 24PMC117

CADL1:

With a text corpus (e.g., a collection of news articles or social media posts), perform the following pre-processing steps using Python libraries (NLTK, SpaCy):

Tokenization, Stemming, Lemmatization, and Stop word removal.

Document the code and the results of each pre-processing step and upload their GitHub link in a padlet.

GitHub Link : <https://github.com/arjuntani/CADL1.git>

Code:

1. Import & Download Resources

```
import nltk
```

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

```
nltk.download('punkt_tab')
```

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

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

```
import spacy
```

```
nlp = spacy.load("en_core_web_sm")
```

- **NLTK** and **spaCy** are two popular NLP libraries.
 - punkt → Needed for sentence/word tokenization.
 - punkt_tab → Supports additional tokenization models.
 - stopwords → List of common words like *is*, *the*, *and*, *of* etc.
 - wordnet → A lexical database required for **lemmatization**.
 - spacy.load("en_core_web_sm") → Loads a pre-trained English NLP pipeline (tokenizer, POS tagger, lemmatizer, stopword list, etc.).
-

2. Sample Dataset

```
corpus = [  
    "The stock market crashed due to global uncertainty.",  
    "Natural Language Processing is a key part of Artificial Intelligence.",  
    "Google releases a new AI model to improve search results.",  
    "The weather today is sunny and pleasant in New York.",  
    "Sports events are being postponed because of heavy rains."  
]
```

👉 A **corpus** is just a collection of text documents.

Here, you created 5 short sentences to test preprocessing.

3. Print Original Corpus

```
print("📌 Original Corpus:")  
  
for i, doc in enumerate(corpus, 1):  
    print(f"{i}. {doc}")
```

👉 Displays each sentence with its index before applying NLP steps.

4. NLTK Preprocessing

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

```
stop_words = set(stopwords.words('english'))
```

```
stemmer = PorterStemmer()
```

```
lemmatizer = WordNetLemmatizer()
```

- word_tokenize → Splits text into words.
- stopwords.words('english') → English stop word list.
- PorterStemmer() → Reduces words to their root form (but not always meaningful).
 - Example: *running* → *run*, *studies* → *studi*

- WordNetLemmatizer() → Uses vocabulary + grammar to reduce to dictionary form.
 - Example: *running* → *run*, *studies* → *study*
-

Processing Each Sentence (NLTK)

for i, doc in enumerate(corpus, 1):

tokens = word_tokenize(doc.lower())

no_stop = [w for w in tokens if w.isalpha() and w not in stop_words]

stemmed = [stemmer.stem(w) for w in no_stop]

lemmatized = [lemmatizer.lemmatize(w) for w in no_stop]

1. doc.lower() → Convert sentence to lowercase.
2. word_tokenize(...) → Breaks into words (tokens).
3. [w for w in tokens if w.isalpha() and w not in stop_words] → Keeps only alphabetic words & removes stopwords.
 - Example: *"the"* → *removed*, *"is"* → *removed*.
4. stemmer.stem(w) → Applies stemming.
5. lemmatizer.lemmatize(w) → Applies lemmatization.

👉 The print statements then show each stage for every sentence.

5. spaCy Preprocessing

for i, doc in enumerate(corpus, 1):

spacy_doc = nlp(doc.lower())

tokens = [token.text for token in spacy_doc]

no_stop = [token.text for token in spacy_doc if not token.is_stop and token.is_alpha]

lemmatized = [token.lemma_ for token in spacy_doc if not token.is_stop and token.is_alpha]

- nlp(doc.lower()) → Passes sentence to spaCy pipeline.
- token.text → Extracts tokens.
- token.is_stop → Checks if word is a stopword.
- token.is_alpha → Ensures only alphabetic tokens (no numbers/punctuations).

- token.lemma_ → Gets the **lemma** (root dictionary form).

👉 SpaCy doesn't have a built-in stemmer because **lemmatization is more accurate**.

Output:

```

===== NLTK Preprocessing =====

Sentence 1: The stock market crashed due to global uncertainty.
★ Tokens: ['the', 'stock', 'market', 'crashed', 'due', 'to', 'global', 'uncertainty', '.']
★ After Stopword Removal: ['stock', 'market', 'crashed', 'due', 'global', 'uncertainty']
★ After Stemming: ['stock', 'market', 'crash', 'due', 'global', 'uncertaini']
★ After Lemmatization: ['stock', 'market', 'crashed', 'due', 'global', 'uncertainty']

Sentence 2: Natural Language Processing is a key part of Artificial Intelligence.
★ Tokens: ['natural', 'language', 'processing', 'is', 'a', 'key', 'part', 'of', 'artificial', 'intelligence', '.']
★ After Stopword Removal: ['natural', 'language', 'processing', 'key', 'part', 'artificial', 'intelligence']
★ After Stemming: ['natur', 'languag', 'process', 'key', 'part', 'artifici', 'intellig']
★ After Lemmatization: ['natural', 'language', 'processing', 'key', 'part', 'artificial', 'intelligence']

Sentence 3: Google releases a new AI model to improve search results.
★ Tokens: ['google', 'releases', 'a', 'new', 'ai', 'model', 'to', 'improve', 'search', 'results', '.']
★ After Stopword Removal: ['google', 'releases', 'new', 'ai', 'model', 'improve', 'search', 'results']
★ After Stemming: ['googl', 'releas', 'new', 'ai', 'model', 'improv', 'search', 'result']
★ After Lemmatization: ['google', 'release', 'new', 'ai', 'model', 'improve', 'search', 'result']

Sentence 4: The weather today is sunny and pleasant in New York.
★ Tokens: ['the', 'weather', 'today', 'is', 'sunny', 'and', 'pleasant', 'in', 'new', 'york', '.']
★ After Stopword Removal: ['weather', 'today', 'sunny', 'pleasant', 'new', 'york']
★ After Stemming: ['weather', 'today', 'sunni', 'pleasant', 'new', 'york']
★ After Lemmatization: ['weather', 'today', 'sunny', 'pleasant', 'new', 'york']

Sentence 5: Sports events are being postponed because of heavy rains.
★ Tokens: ['sports', 'events', 'are', 'being', 'postponed', 'because', 'of', 'heavy', 'rains', '.']
★ After Stopword Removal: ['sports', 'events', 'postponed', 'heavy', 'rains']
★ After Stemming: ['sport', 'event', 'postpon', 'heavi', 'rain']
★ After Lemmatization: ['sport', 'event', 'postponed', 'heavy', 'rain']

===== spaCy Preprocessing =====

Sentence 1: The stock market crashed due to global uncertainty.
★ Tokens: ['the', 'stock', 'market', 'crashed', 'due', 'to', 'global', 'uncertainty', '.']
★ After Stopword Removal: ['stock', 'market', 'crashed', 'global', 'uncertainty']
★ After Lemmatization: ['stock', 'market', 'crash', 'global', 'uncertainty']

Sentence 2: Natural Language Processing is a key part of Artificial Intelligence.
★ Tokens: ['natural', 'language', 'processing', 'is', 'a', 'key', 'part', 'of', 'artificial', 'intelligence', '.']
★ After Stopword Removal: ['natural', 'language', 'processing', 'key', 'artificial', 'intelligence']
★ After Lemmatization: ['natural', 'language', 'processing', 'key', 'artificial', 'intelligence']

Sentence 3: Google releases a new AI model to improve search results.
★ Tokens: ['google', 'releases', 'a', 'new', 'ai', 'model', 'to', 'improve', 'search', 'results', '.']
★ After Stopword Removal: ['google', 'releases', 'new', 'ai', 'model', 'improve', 'search', 'results']
★ After Lemmatization: ['google', 'release', 'new', 'ai', 'model', 'improve', 'search', 'result']

Sentence 4: The weather today is sunny and pleasant in New York.
★ Tokens: ['the', 'weather', 'today', 'is', 'sunny', 'and', 'pleasant', 'in', 'new', 'york', '.']
★ After Stopword Removal: ['weather', 'today', 'sunny', 'pleasant', 'new', 'york']
★ After Lemmatization: ['weather', 'today', 'sunny', 'pleasant', 'new', 'york']

Sentence 5: Sports events are being postponed because of heavy rains.
★ Tokens: ['sports', 'events', 'are', 'being', 'postponed', 'because', 'of', 'heavy', 'rains', '.']
★ After Stopword Removal: ['sports', 'events', 'postponed', 'heavy', 'rains']
★ After Lemmatization: ['sport', 'event', 'postpone', 'heavy', 'rain']

```