Name: **B. SAI CHARAN**

Roll No: **2203A51L72**

Batch No: **21CSBTB12**

ASSIGNMENT - 2

Q: **Preparation of datasets by applying ambiguity removal, segmentation, stemming.**

A: In Natural Language Processing (NLP), the quality of the input data significantly impacts the performance of the models. Preparing datasets by applying techniques such as ambiguity removal, segmentation, and stemming is crucial to ensure that the data is clean, relevant, and suitable for analysis or model training.

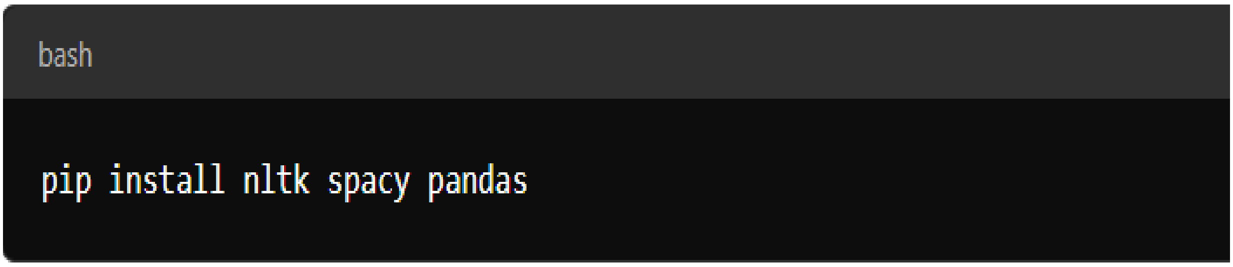
Materials

- Computer with Python installed
- Text dataset (e.g., .txt files or CSV files)
- Python libraries: nltk, spaCy, pandas, re.

Experiment Steps

1. Setup Environment

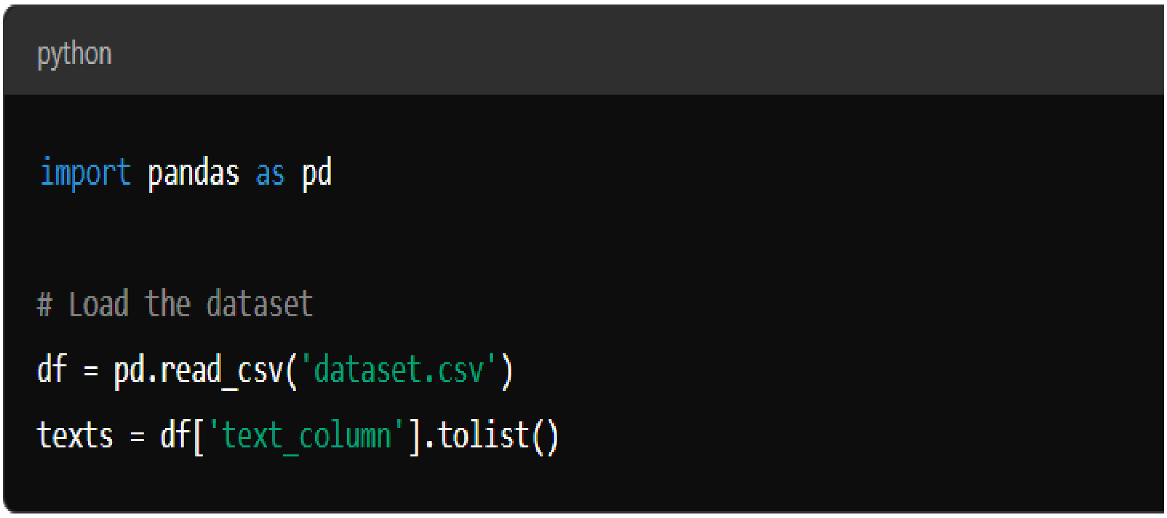
- Ensure Python and necessary libraries are installed



```bash
pip install nltk spacy pandas
```

2. Load Dataset

- Load your dataset into a DataFrame or a list.
- Example (CSV file):

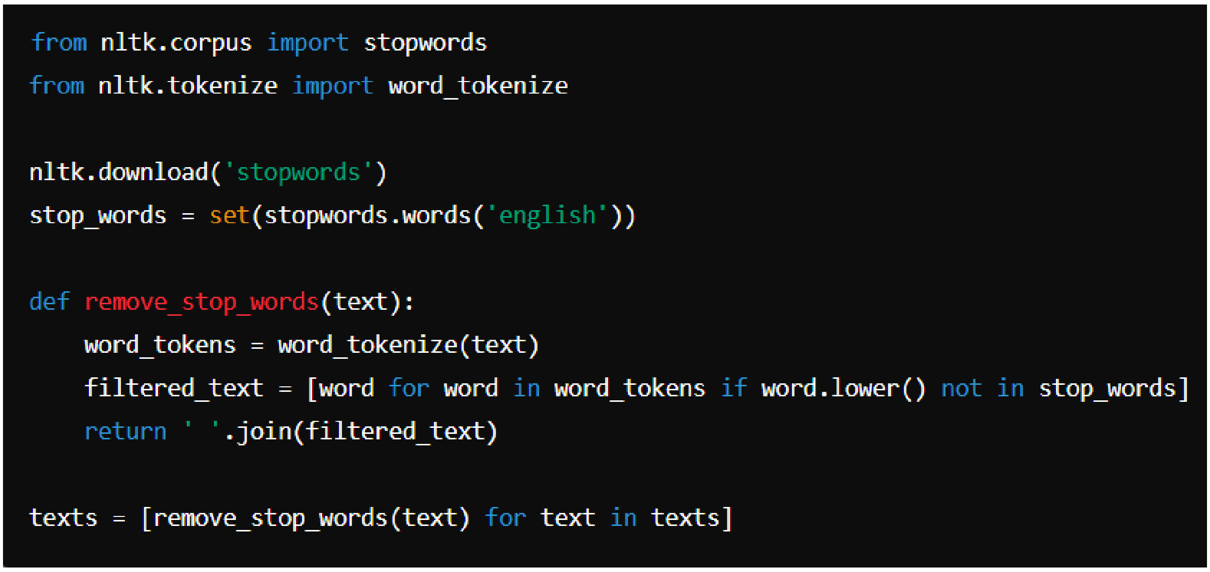


python

```python
import pandas as pd

# Load the dataset
df = pd.read_csv('dataset.csv')
texts = df['text_column'].tolist()
```

3. Ambiguity Removal

- Remove Stop Words:



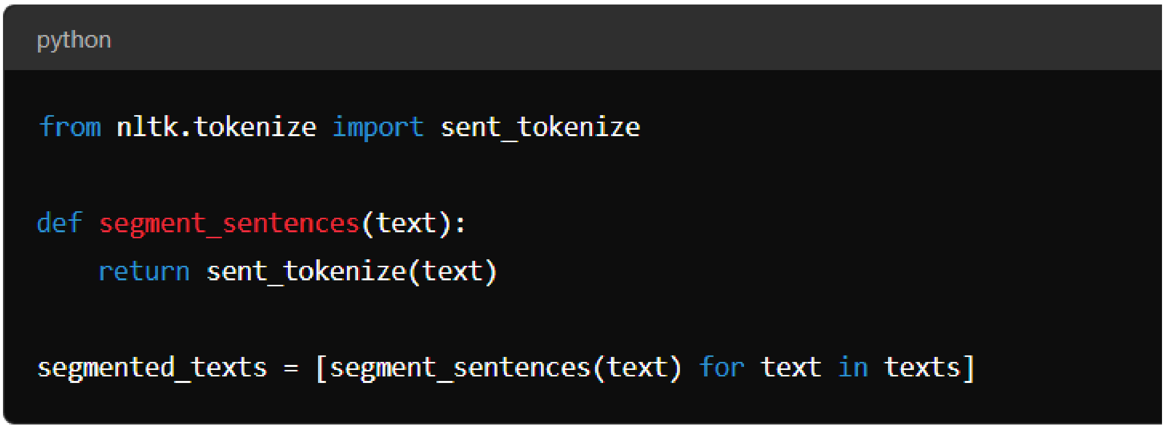
```python
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

nltk.download('stopwords')
stop_words = set(stopwords.words('english'))

def remove_stop_words(text):
    word_tokens = word_tokenize(text)
    filtered_text = [word for word in word_tokens if word.lower() not in stop_words]
    return ' '.join(filtered_text)

texts = [remove_stop_words(text) for text in texts]
```

Handle Homonyms and Polysemy:

- Consider using a named entity recognizer or disambiguation techniques based on context.

4. Segmentation

- **Sentence Segmentation:**



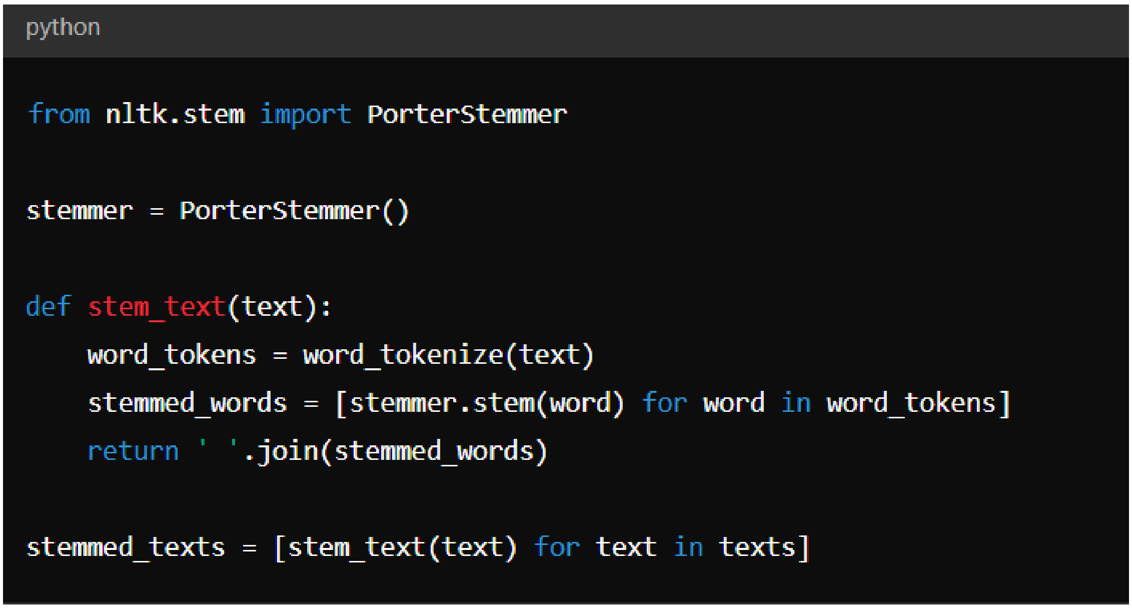
```python
from nltk.tokenize import sent_tokenize

def segment_sentences(text):
    return sent_tokenize(text)

segmented_texts = [segment_sentences(text) for text in texts]
```

Word Segmentation (if necessary):

- For languages where word segmentation is needed (e.g., Chinese), use appropriate libraries like ‘`jieba`’.

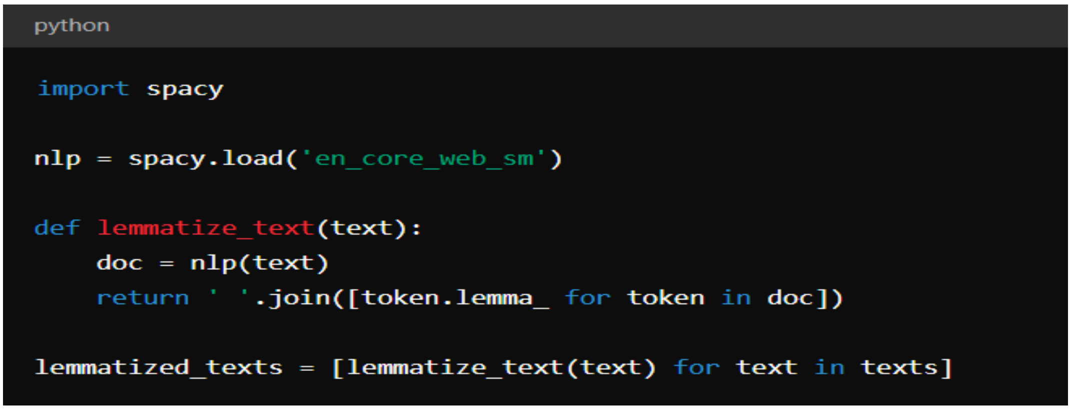
5. Stemming

- **Apply Stemming:**



```python
from nltk.stem import PorterStemmer

stemmer = PorterStemmer()

def stem_text(text):
    word_tokens = word_tokenize(text)
    stemmed_words = [stemmer.stem(word) for word in word_tokens]
    return ' '.join(stemmed_words)

stemmed_texts = [stem_text(text) for text in texts]
```

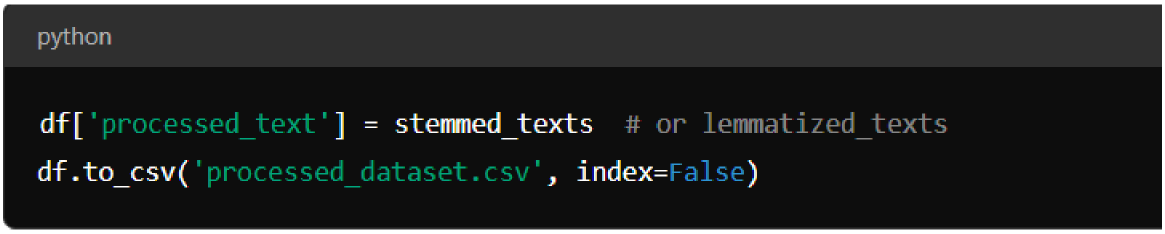
- **Alternative: Lemmatization:**
    - You can use `spaCy` for lemmatization:



```python
import spacy

nlp = spacy.load('en_core_web_sm')

def lemmatize_text(text):
    doc = nlp(text)
    return ' '.join([token.lemma_ for token in doc])

lemmatized_texts = [lemmatize_text(text) for text in texts]
```

6. Save Processed Data

- Save the preprocessed data to a new file or DataFrame



```python
df['processed_text'] = stemmed_texts  # or lemmatized_texts
df.to_csv('processed_dataset.csv', index=False)
```