Write a program in Python to implement text pre-processing using NLTK and spaCy.

import nltk

nltk.download('punkt')

import nltk

nltk.download('stopwords')

import nltk

from nltk.tokenize import word\_tokenize

from nltk.corpus import stopwords

from nltk.stem import PorterStemmer

import spacy

# Sample text

text = "John owns a gold watch and loves to watch the sunrise every morning."

text

# Tokenization

tokens = word\_tokenize(text)

tokens

# Stop-word Removal

stop\_words = set(stopwords.words('english'))

filtered\_tokens = [word for word in tokens if word.lower() not in stop\_words]

filtered\_tokens

# Stemming

ps = PorterStemmer()

stemmed\_tokens = [ps.stem(word) for word in filtered\_tokens]

stemmed\_tokens

# Lemmatization with spaCy

nlp = spacy.load("en\_core\_web\_sm")

doc = nlp(text)

lemmatized\_tokens = [token.lemma\_ for token in doc]

lemmatized\_tokens

Write a program in Python to implement a Part-of-Speech (POS) Tagger using supervised machine learning algorithms such as the Hidden Markov Model (HMM) or the Maximum Entropy Markov Model (MEMM).

import nltk

from nltk.tag import hmm

from sklearn.model\_selection import train\_test\_split

from nltk.corpus import Treebank

import nltk

nltk.download('treebank')

# Load tagged sentences from the Treebank corpus

data = treebank.tagged\_sents()

# Split data into training and testing sets

train\_data, test\_data = train\_test\_split(data, test\_size=0.2)

# Initialize the HMM trainer

trainer = hmm.HiddenMarkovModelTrainer()

# Train the HMM tagger

hmm\_tagger = trainer.train(train\_data)

# Define a sample sentence to tag

sentence = ["This", "is", "a", "sample", "sentence", "for", "testing", "."]

# Tag the sample sentence

tagged\_sentence = hmm\_tagger.tag(sentence)

# Display the tagged output

print("Tagged Sentence:", tagged\_sentence)

In this output:

DT :- represents a determiner.

VBZ:- represents a third-person singular verb.

NN :- represents a noun.

IN :- represents a preposition or subordinating conjunction.