```
# Install NLTK and spaCy
!pip install nltk spacy
!python -m spacy download en_core_web_sm

# Import required modules
import nltk
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer
import spacy

# Download NLTK data (if not already downloaded)
nltk.download('punkt')
nltk.download('wordnet')
nltk.download('punkt_tab') # Added to download the missing resource

# Load spaCy model
nlp = spacy.load('en_core_web_sm')

print("Libraries installed and imported successfully.")
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.12/dist-packages (3.9.1)
Requirement already satisfied: spacy in /usr/local/lib/python3.12/dist-packages (3.8.11)
Requirement already satisfied: click in /usr/local/lib/python3.12/dist-packages (from nltk) (8.3.1
Requirement already satisfied: joblib in /usr/local/lib/python3.12/dist-packages (from nltk) (1.5.
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.12/dist-packages (from nl
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from nltk) (4.67.1
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.12/dist-packa
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.12/dist-packa
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.12/dist-package
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.12/dist-packages (frc
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.12/dist-packages (f
Requirement already satisfied: thinc<8.4.0,>=8.3.4 in /usr/local/lib/python3.12/dist-packages (frc
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.12/dist-packages (fr
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.12/dist-packages (frc
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: weasel<0.5.0,>=0.4.2 in /usr/local/lib/python3.12/dist-packages (fr
Requirement already satisfied: typer-slim<1.0.0,>=0.3.0 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: numpy>=1.19.0 in /usr/local/lib/python3.12/dist-packages (from spac
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.12/d
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.1
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from spacy)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from sp
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (
Requirement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (f
Requirement already satisfied: typing-extensions>=4.14.1 in /usr/local/lib/python3.12/dist-package
Requirement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from reque
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from
Requirement already satisfied: blis<1.4.0,>=1.3.0 in /usr/local/lib/python3.12/dist-packages (from
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.12/dist-packag
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from ji
Requirement already satisfied: wrapt in /usr/local/lib/python3.12/dist-packages (from smart-open<8
Collecting en-core-web-sm==3.8.0
  Downloading https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-3.8.0/en_
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.8/12.8 MB 103.9 MB/s eta 0:00:00
✓ Download and installation successful
```

```
You can now load the package via spacy.load('en_core_web_sm')
⚠ Restart to reload dependencies
If you are in a Jupyter or Colab notebook, you may need to restart Python in
order to load all the package's dependencies. You can do this by selecting the
'Restart kernel' or 'Restart runtime' option.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
Libraries installed and imported successfully.
```

```
medical_text = """A patient presented with symptoms indicative of seasonal allergies, including r

print("Medical text loaded successfully.")
```

```
Medical text loaded successfully.
```

```
# Sentence Tokenization (NLTK)
sentences_nltk = nltk.sent_tokenize(medical_text)
print("NLTK Sentence Tokenization:")
for i, sentence in enumerate(sentences_nltk):
    print(f"Sentence {i+1}: {sentence}")
```

```
NLTK Sentence Tokenization:
Sentence 1: A patient presented with symptoms indicative of seasonal allergies, including rhinorrh
Sentence 2: Physical examination revealed clear lung sounds and no signs of acute distress.
Sentence 3: The patient's history included a known sensitivity to pollen.
Sentence 4: Treatment involved prescribing a non-sedating antihistamine and advising on environmen
Sentence 5: Follow-up is scheduled in two weeks to assess symptom improvement and adjust medicatic
```

```
# Word Tokenization (NLTK)
words_nltk = [word_tokenize(sentence) for sentence in sentences_nltk]
print("\nNLTK Word Tokenization:")
for i, sentence_words in enumerate(words_nltk):
    print(f"Sentence {i+1}: {sentence_words}")

# Word Tokenization (spaCy)
doc = nlp(medical_text)
words_spacy = [[token.text for token in sent] for sent in doc.sents]
print("\nspaCy Word Tokenization:")
for i, sentence_words in enumerate(words_spacy):
    print(f"Sentence {i+1}: {sentence_words}")
```

```

NLTK Word Tokenization:
Sentence 1: ['A', 'patient', 'presented', 'with', 'symptoms', 'indicative', 'of', 'seasonal', 'all
Sentence 2: ['Physical', 'examination', 'revealed', 'clear', 'lung', 'sounds', 'and', 'no', 'signs
Sentence 3: ['The', 'patient', "'s", 'history', 'included', 'a', 'known', 'sensitivity', 'to', 'pc
Sentence 4: ['Treatment', 'involved', 'prescribing', 'a', 'non-sedating', 'antihistamine', 'and',
Sentence 5: ['Follow-up', 'is', 'scheduled', 'in', 'two', 'weeks', 'to', 'assess', 'symptom', 'imp

spaCy Word Tokenization:
Sentence 1: ['A', 'patient', 'presented', 'with', 'symptoms', 'indicative', 'of', 'seasonal', 'all
Sentence 2: ['Physical', 'examination', 'revealed', 'clear', 'lung', 'sounds', 'and', 'no', 'signs
Sentence 3: ['The', 'patient', "'s", 'history', 'included', 'a', 'known', 'sensitivity', 'to', 'pc
Sentence 4: ['Treatment', 'involved', 'prescribing', 'a', 'non', '-', 'sedating', 'antihistamine',
Sentence 5: ['Follow', '-', 'up', 'is', 'scheduled', 'in', 'two', 'weeks', 'to', 'assess', 'symptc
```

```
# Apply Stemming (Porter Stemmer)
porter_stemmer = PorterStemmer()

stemmed_words_nltk = []
for sentence_words in words_nltk:
    stemmed_sentence = [porter_stemmer.stem(word) for word in sentence_words]
    stemmed_words_nltk.append(stemmed_sentence)

print("\nPorter Stemming (NLTK):")
for i, sentence_words in enumerate(stemmed_words_nltk):
    print(f"Sentence {i+1}: {sentence_words}")
```

```
Porter Stemming (NLTK):
Sentence 1: ['a', 'patient', 'present', 'with', 'symptom', 'indic', 'of', 'season', 'allergi', ','
Sentence 2: ['physic', 'examin', 'reveal', 'clear', 'lung', 'sound', 'and', 'no', 'sign', 'of', 'a
Sentence 3: ['the', 'patient', "'s", 'histori', 'includ', 'a', 'known', 'sensit', 'to', 'pollen',
Sentence 4: ['treatment', 'involv', 'prescrib', 'a', 'non-sed', 'antihistamin', 'and', 'advis', 'c
Sentence 5: ['follow-up', 'is', 'schedul', 'in', 'two', 'week', 'to', 'assess', 'symptom', 'improv
```

```
# Apply Lemmatization (NLTK)
lemmatizer = WordNetLemmatizer()

lemmatized_words_nltk = []
for sentence_words in words_nltk:
    lemmatized_sentence = [lemmatizer.lemmatize(word) for word in sentence_words]
    lemmatized_words_nltk.append(lemmatized_sentence)

print("\nNLTK Lemmatization:")
for i, sentence_words in enumerate(lemmatized_words_nltk):
    print(f"Sentence {i+1}: {sentence_words}")

# Apply Lemmatization (spaCy)
lemmatized_words_spacy = [[token.lemma_ for token in sent] for sent in doc.sents]

print("\nspaCy Lemmatization:")
for i, sentence_words in enumerate(lemmatized_words_spacy):
    print(f"Sentence {i+1}: {sentence_words}")
```

```
NLTK Lemmatization:
Sentence 1: ['A', 'patient', 'presented', 'with', 'symptom', 'indicative', 'of', 'seasonal', 'alle
Sentence 2: ['Physical', 'examination', 'revealed', 'clear', 'lung', 'sound', 'and', 'no', 'sign',
Sentence 3: ['The', 'patient', "'s", 'history', 'included', 'a', 'known', 'sensitivity', 'to', 'pc
Sentence 4: ['Treatment', 'involved', 'prescribing', 'a', 'non-sedating', 'antihistamine', 'and',
Sentence 5: ['Follow-up', 'is', 'scheduled', 'in', 'two', 'week', 'to', 'assess', 'symptom', 'impr

spaCy Lemmatization:
Sentence 1: ['a', 'patient', 'present', 'with', 'symptom', 'indicative', 'of', 'seasonal', 'allerg
Sentence 2: ['physical', 'examination', 'reveal', 'clear', 'lung', 'sound', 'and', 'no', 'sign', '
Sentence 3: ['the', 'patient', "'s", 'history', 'include', 'a', 'know', 'sensitivity', 'to', 'poll
Sentence 4: ['treatment', 'involve', 'prescribe', 'a', 'non', '-', 'sedating', 'antihistamine', 'a
Sentence 5: ['follow', '-', 'up', 'be', 'schedule', 'in', 'two', 'week', 'to', 'assess', 'symptom'
```

```
print("\n--- Comparison of Original, Stemmed, and Lemmatized Words (Sentence 1) ---")
print(f"{"Original (NLTK)":<20} {"Stemmed (NLTK)":<20} {"Lemmatized (NLTK)":<20} {"Lemmatized (sp
print("-" * 80)

# Take the first sentence's NLTK tokenized words as reference
# For the first sentence, NLTK and spaCy tokenization outputs were identical,
```

```
# allowing for a direct word-by-word comparison across all forms.
for original_word, stemmed_word_nltk, lemma_word_nltk, lemma_word_spacy in zip(
    words_nltk[0],
    stemmed_words_nltk[0],
    lemmatized_words_nltk[0],
    lemmatized_words_spacy[0]
):
    print(f"{original_word:<20} {stemmed_word_nltk:<20} {lemma_word_nltk:<20} {lemma_word_spacy:<

print("\nObservations:")
print("- Stemming often results in truncated words that are not actual dictionary words (e.g., 'a
print("- NLTK Lemmatization generally provides correct base forms, but sometimes requires Part-of
print("- spaCy Lemmatization is often more sophisticated, providing correct base forms, handling
print("- This comparison highlights why lemmatization is generally preferred for applications whe
```

```
--- Comparison of Original, Stemmed, and Lemmatized Words (Sentence 1) ---
Original (NLTK)      Stemmed (NLTK)       Lemmatized (NLTK)    Lemmatized (spaCy)
--------------------------------------------------------------------------------
A                    a                    A                    a
patient              patient              patient              patient
presented            present              presented            present
with                 with                 with                 with
symptoms             symptom              symptom              symptom
indicative           indic                indicative           indicative
of                   of                   of                   of
seasonal             season               seasonal             seasonal
allergies            allergi              allergy              allergy
,                    ,                    ,                    ,
including            includ               including            include
rhinorrhea           rhinorrhea           rhinorrhea           rhinorrhea
,                    ,                    ,                    ,
nasal                nasal                nasal                nasal
congestion           congest              congestion           congestion
,                    ,                    ,                    ,
and                  and                  and                  and
mild                 mild                 mild                 mild
ocular               ocular               ocular               ocular
pruritus             pruritu              pruritus             pruritus
.                    .                    .                    .

Observations:
- Stemming often results in truncated words that are not actual dictionary words (e.g., 'allergies
- NLTK Lemmatization generally provides correct base forms, but sometimes requires Part-of-Speech
- spaCy Lemmatization is often more sophisticated, providing correct base forms, handling verb ten
- This comparison highlights why lemmatization is generally preferred for applications where seman
```