```
!pip install NLTK spacy
import nltk
import spacy
nltk.download('punkt')
nltk.download('wordnet')
nltk.download('averaged_perceptron_tagger')
nltk.download('stopwords')
nltk.download('punkt_tab') # Added to download the missing resource
nltk.download('averaged_perceptron_tagger_eng') # Added to download the missing resource for NLTK lemmatization
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer, WordNetLemmatizer
from nltk.corpus import stopwords, wordnet # Added wordnet import

# Load the English spaCy model. You might need to download it first if not already present.
try:
    nlp = spacy.load('en_core_web_sm')
except OSError:
    print('Downloading en_core_web_sm model for spaCy...')
    spacy.cli.download('en_core_web_sm')
    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) (1.5.3)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.12/dist-packages (from NLTK) (2025.11.3)
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-packages (from spacy) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.0.15)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.13)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.0.12)
Requirement already satisfied: thinc<8.4.0,>=8.3.4 in /usr/local/lib/python3.12/dist-packages (from spacy) (8.3.10)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.1.3)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.5.2)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.10)
Requirement already satisfied: weasel<0.5.0,>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (0.4.3)
Requirement already satisfied: typer-slim<1.0.0,>=0.3.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (0.20.0)
Requirement already satisfied: numpy>=1.19.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.2)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.32.4)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.12.3)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.1.6)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from spacy) (75.2.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (25.0)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy) (0.7.0)
Requirement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy) (2.41.4)
Requirement already satisfied: typing-extensions>=4.14.1 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy) (4.15.0)
Requirement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy) (0.4.2)
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (3.4.4)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (3.11)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (2.5.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (2025.11.12)
Requirement already satisfied: blis<1.4.0,>=1.3.0 in /usr/local/lib/python3.12/dist-packages (from thinc<8.4.0,>=8.3.4->spacy) (1.3.3)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.12/dist-packages (from thinc<8.4.0,>=8.3.4->spacy) (0.1.5)
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.12/dist-packages (from weasel<0.5.0,>=0.4.2->spacy) (0.23.0)
```

```
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.12/dist-packages (from weasel<0.5.0,>=0.4.2->spacy) (7.5.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->spacy) (3.0.3)
Requirement already satisfied: wrapt in /usr/local/lib/python3.12/dist-packages (from smart-open<8.0.0,>=5.2.1->weasel<0.5.0,>=0.4.2->spacy) (2.0.1)
[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 averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger_eng to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger_eng is already up-to-
[nltk_data]       date!
Libraries installed and imported successfully.
```

```
medical_text = """
Diabetes is a chronic disease that affects how the body processes blood sugar.
If untreated, diabetes may cause heart disease, kidney failure, nerve damage and vision problems.
Early diagnosis and proper treatment help improve patient outcomes
"""
print("Medical text updated successfully.")
```

```
Medical text updated successfully.
```

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

```
Sentence Tokenization (NLTK):
Sentence 1: A 45-year-old male presented with a two-week history of persistent cough, shortness of breath, and mild fever.
Sentence 2: Physical examination revealed crackles in the lower right lung field.
Sentence 3: Chest X-ray showed a consolidation consistent with pneumonia.
Sentence 4: Laboratory tests indicated elevated white blood cell count.
Sentence 5: Treatment initiated with azithromycin and supportive care resulted in gradual improvement of symptoms over five days.
Sentence 6: Follow-up advised in two weeks.
```

```
# Word Tokenization with NLTK
nltk_tokens = word_tokenize(medical_text)
print("\nWord Tokenization (NLTK):")
print(nltk_tokens)

# Word Tokenization with spaCy
spacy_doc = nlp(medical_text)
spacy_tokens = [token.text for token in spacy_doc]
print("\nWord Tokenization (spaCy):")
print(spacy_tokens)
```

```

Word Tokenization (NLTK):
['A', '45-year-old', 'male', 'presented', 'with', 'a', 'two-week', 'history', 'of', 'persistent', 'cough', ',', 'shortness', 'of', 'breath', ',', 'and', 'mild', 'fever'
```

```
Word Tokenization (spaCy):
['A', '45', '-', 'year', '-', 'old', 'male', 'presented', 'with', 'a', 'two', '-', 'week', 'history', 'of', 'persistent', 'cough', ',', 'shortness', 'of', 'breath', ','
```

```
# Initialize Porter Stemmer
porter_stemmer = PorterStemmer()

# Apply stemming to NLTK tokens
stemmed_words = [porter_stemmer.stem(word) for word in nltk_tokens]

print("\nStemming (Porter Stemmer):")
print(stemmed_words)
```

```
Stemming (Porter Stemmer):
['a', '45-year-old', 'male', 'present', 'with', 'a', 'two-week', 'histori', 'of', 'persist', 'cough', ',', 'short', 'of', 'breath', ',', 'and', 'mild', 'fever', '.', 'p
```

```
# Initialize WordNet Lemmatizer
wordnet_lemmatizer = WordNetLemmatizer()

# Function to convert NLTK POS tag to WordNet POS tag for lemmatization
def get_wordnet_pos(word):
    tag = nltk.pos_tag([word])[0][1][0].upper()
    tag_dict = {"J": wordnet.ADJ, "N": wordnet.NOUN, "V": wordnet.VERB, "R": wordnet.ADV}
    return tag_dict.get(tag, wordnet.NOUN)

# Apply Lemmatization with NLTK
# We filter out non-alphabetic tokens to avoid errors with lemmatizer
nltk_lemmas = [wordnet_lemmatizer.lemmatize(word, get_wordnet_pos(word)) for word in nltk_tokens if word.isalpha()]
print("\nLemmatization (NLTK):")
print(nltk_lemmas)

# Apply Lemmatization with spaCy
spacy_lemmas = [token.lemma_ for token in spacy_doc]
print("\nLemmatization (spaCy):")
print(spacy_lemmas)
```

```
Lemmatization (NLTK):
['A', 'male', 'present', 'with', 'a', 'history', 'of', 'persistent', 'cough', 'shortness', 'of', 'breath', 'and', 'mild', 'fever', 'Physical', 'examination', 'reveal',

Lemmatization (spaCy):
['a', '45', '-', 'year', '-', 'old', 'male', 'present', 'with', 'a', 'two', '-', 'week', 'history', 'of', 'persistent', 'cough', ',', 'shortness', 'of', 'breath', ',',
```

```
# Select a few representative words to compare
comparison_words = [
    "presented", "history", "shortness", "revealed", "crackles",
    "indicated", "elevated", "improvement", "symptoms", "days", "weeks"
]

print("Comparison of Original Words, Stemmed Words (NLTK), and Lemmatized Words (NLTK & spaCy):\n")
print(f"{'Original Word':<15} {'NLTK Stem':<15} {'NLTK Lemma':<15} {'spaCy Lemma':<15}")
print("-" * 65)
```

```
for word in comparison_words:
    # NLTK Stemming
    stemmed_word = porter_stemmer.stem(word)

    # NLTK Lemmatization
    nltk_lemma_val = word # Default to original
    if word.isalpha(): # Only attempt POS tagging and lemmatization for alphabetic words
        try:
            nltk_lemma_val = wordnet_lemmatizer.lemmatize(word, get_wordnet_pos(word))
        except:
            # Fallback if POS tagging fails for some reason (e.g., rare word)
            nltk_lemma_val = wordnet_lemmatizer.lemmatize(word)

    # spaCy Lemmatization (process the individual word to get its lemma)
    spacy_token = nlp(word.lower())[0] # Process as lowercase to handle case variations
    spacy_lemma_val = spacy_token.lemma_

    print(f"{word:<15} {stemmed_word:<15} {nltk_lemma_val:<15} {spacy_lemma_val:<15}")

print("\n\nObservations:")
print("- **Stemming** (e.g., Porter Stemmer) is a heuristic process that chops off suffixes to get to a 'root' form. This is fast but often results in forms that are n
print("- **Lemmatization** (NLTK and spaCy) aims to return the base or dictionary form of a word (the 'lemma'), ensuring it is a real, grammatically correct word. For
print("- In medical contexts, preserving the meaning and grammatical correctness of terms is crucial. Therefore, **lemmatization is generally preferred over stemming**
print("- **spaCy's lemmatizer**, leveraging statistical models and a deeper understanding of language structure, often produces more accurate and contextually appropri
```

```
Comparison of Original Words, Stemmed Words (NLTK), and Lemmatized Words (NLTK & spaCy):

Original Word   NLTK Stem       NLTK Lemma      spaCy Lemma
--------------------------------------------------------------------
presented       present         present         present
history         histori         history         history
shortness       short           shortness       shortness
revealed        reveal          reveal          reveal
crackles        crackl          crackle         crackle
indicated       indic           indicate        indicate
elevated        elev            elevate         elevate
improvement     improv          improvement     improvement
symptoms        symptom         symptom         symptom
days            day             day             day
weeks           week            week            week


Observations:
- **Stemming** (e.g., Porter Stemmer) is a heuristic process that chops off suffixes to get to a 'root' form. This is fast but often results in forms that are not actua
- **Lemmatization** (NLTK and spaCy) aims to return the base or dictionary form of a word (the 'lemma'), ensuring it is a real, grammatically correct word. For example,
- In medical contexts, preserving the meaning and grammatical correctness of terms is crucial. Therefore, **lemmatization is generally preferred over stemming**, as it
- **spaCy's lemmatizer**, leveraging statistical models and a deeper understanding of language structure, often produces more accurate and contextually appropriate lemm
```

```
Start coding or generate with AI.
```