```
import nltk
import spacy
```

```
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."""
```

```
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."""
import nltk
nltk.download('punkt_tab') # 'punkt_tab' is the correct resource for sent_tokenize, as indicated by the error message

sentences = nltk.sent_tokenize(medical_text)

print("Sentence Tokenization with NLTK:")
for i, sent in enumerate(sentences):
    print(f"Sentence {i+1}: {sent}")
```

```
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
Sentence Tokenization with NLTK:
Sentence 1: Diabetes is a chronic disease that affects how the body processes blood sugar.
Sentence 2: If untreated, diabetes may cause heart disease, kidney failure, nerve damage and vision problems.
Sentence 3: Early diagnosis and proper treatment help improve patient outcomes.
```

```
import nltk
import spacy

# Assuming 'sentences' variable is already available from previous execution
# If not, you might need to re-run the sentence tokenization cell or define medical_text and sentences here.

# NLTK Word Tokenization
print("\nWord Tokenization with NLTK:")
for i, sent in enumerate(sentences):
    words = nltk.word_tokenize(sent)
    print(f"Sentence {i+1} words: {words}")

# spaCy Word Tokenization
# Load the English language model for spaCy
try:
    nlp = spacy.load("en_core_web_sm")
except OSError:
    print("Downloading spaCy 'en_core_web_sm' model...")
    spacy.cli.download("en_core_web_sm")
    nlp = spacy.load("en_core_web_sm")

print("\nWord Tokenization with spaCy:")
for i, sent in enumerate(sentences):
    doc = nlp(sent)
    words = [token.text for token in doc]
    print(f"Sentence {i+1} words: {words}")
```

```
Word Tokenization with NLTK:
Sentence 1 words: ['Diabetes', 'is', 'a', 'chronic', 'disease', 'that', 'affects', 'how', 'the', 'body', 'processes', 'blooc
Sentence 2 words: ['If', 'untreated', ',', 'diabetes', 'may', 'cause', 'heart', 'disease', ',', 'kidney', 'failure', ',', 'r
Sentence 3 words: ['Early', 'diagnosis', 'and', 'proper', 'treatment', 'help', 'improve', 'patient', 'outcomes', '.']

Word Tokenization with spaCy:
Sentence 1 words: ['Diabetes', 'is', 'a', 'chronic', 'disease', 'that', 'affects', 'how', 'the', 'body', 'processes', 'blooc
Sentence 2 words: ['If', 'untreated', ',', 'diabetes', 'may', 'cause', 'heart', 'disease', ',', 'kidney', 'failure', ',', 'r
Sentence 3 words: ['Early', 'diagnosis', 'and', 'proper', 'treatment', 'help', 'improve', 'patient', 'outcomes', '.']
```

```
from nltk.stem import PorterStemmer
import nltk

# Initialize Porter Stemmer
porter = PorterStemmer()

print("\nApplying Porter Stemming:")
for i, sent in enumerate(sentences):
    words = nltk.word_tokenize(sent)
    stemmed_words = [porter.stem(word) for word in words]
    print(f"Sentence {i+1} stemmed words: {stemmed_words}")
```

```
Applying Porter Stemming:
Sentence 1 stemmed words: ['diabet', 'is', 'a', 'chronic', 'diseas', 'that', 'affect', 'how', 'the', 'bodi', 'process', 'blc
```

```
Sentence 2 stemmed words: ['if', 'untreat', ',', 'diabet', 'may', 'caus', 'heart', 'diseas', ',', 'kidney', 'failur', ',', '
Sentence 3 stemmed words: ['earli', 'diagnosi', 'and', 'proper', 'treatment', 'help', 'improv', 'patient', 'outcom', '.']
```

Start coding or generate with AI.

```
import nltk
from nltk.stem import WordNetLemmatizer
import spacy

# Download necessary NLTK resources for WordNetLemmatizer if not already present
try:
    nltk.data.find('corpora/wordnet')
except LookupError:
    print("Downloading NLTK 'wordnet' resource...")
    nltk.download('wordnet')
try:
    nltk.data.find('corpora/omw-1.4')
except LookupError:
    print("Downloading NLTK 'omw-1.4' resource...")
    nltk.download('omw-1.4')

# Initialize WordNet Lemmatizer
lemmatizer = WordNetLemmatizer()

print("\nApplying NLTK WordNet Lemmatization:")
for i, sent in enumerate(sentences):
    words = nltk.word_tokenize(sent)
    lemmatized_words = [lemmatizer.lemmatize(word) for word in words]
    print(f"Sentence {i+1} lemmatized words: {lemmatized_words}")

# spaCy Word Lemmatization
# Load the English language model for spaCy
try:
    nlp = spacy.load("en_core_web_sm")
except OSError:
    print("Downloading spaCy 'en_core_web_sm' model...")
    spacy.cli.download("en_core_web_sm")
    nlp = spacy.load("en_core_web_sm")

print("\nApplying spaCy Lemmatization:")
for i, sent in enumerate(sentences):
    doc = nlp(sent)
    lemmatized_words = [token.lemma_ for token in doc]
    print(f"Sentence {i+1} lemmatized words: {lemmatized_words}")
```

```
Downloading NLTK 'wordnet' resource...
Downloading NLTK 'omw-1.4' resource...
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...

Applying NLTK WordNet Lemmatization:
Sentence 1 lemmatized words: ['Diabetes', 'is', 'a', 'chronic', 'disease', 'that', 'affect', 'how', 'the', 'body', 'process'
Sentence 2 lemmatized words: ['If', 'untreated', ',', 'diabetes', 'may', 'cause', 'heart', 'disease', ',', 'kidney', 'failur
Sentence 3 lemmatized words: ['Early', 'diagnosis', 'and', 'proper', 'treatment', 'help', 'improve', 'patient', 'outcome', '

Applying spaCy Lemmatization:
Sentence 1 lemmatized words: ['diabetes', 'be', 'a', 'chronic', 'disease', 'that', 'affect', 'how', 'the', 'body', 'process'
Sentence 2 lemmatized words: ['if', 'untreate', ',', 'diabete', 'may', 'cause', 'heart', 'disease', ',', 'kidney', 'failure'
Sentence 3 lemmatized words: ['early', 'diagnosis', 'and', 'proper', 'treatment', 'help', 'improve', 'patient', 'outcome', '
```

```
import nltk
from nltk.stem import PorterStemmer, WordNetLemmatizer
import spacy

# Download NLTK resources if not already present
nltk.download('punkt', quiet=True)
nltk.download('wordnet', quiet=True)
nltk.download('omw-1.4', quiet=True)

# Initialize stemmer and lemmatizer
porter = PorterStemmer()
wordnet_lemmatizer = WordNetLemmatizer()

# Load spaCy model
try:
    nlp = spacy.load("en_core_web_sm")
except OSError:
    print("Downloading spaCy 'en_core_web_sm' model...")
    spacy.cli.download("en_core_web_sm")
    nlp = spacy.load("en_core_web_sm")
```

```
print("\n--- Comparison: Original vs. Stemmed vs. Lemmatized ---")
print(f"{'Original Word':<18} | {'Porter Stem':<15} | {'NLTK Lemma':<15} | {'spaCy Lemma':<15}")
print("-" * 70)

# Iterate through sentences and words for comparison
for sent in sentences:
    doc_spacy = nlp(sent) # Process sentence with spaCy once
    for word_nltk, token_spacy in zip(nltk.word_tokenize(sent), doc_spacy):
        original = word_nltk
        stemmed = porter.stem(original)
        nltk_lemma = wordnet_lemmatizer.lemmatize(original)
        spacy_lemma = token_spacy.lemma_ # Get lemma directly from spaCy token

        print(f"{original:<18} | {stemmed:<15} | {nltk_lemma:<15} | {spacy_lemma:<15}")
    print("-" * 70) # Separator for each sentence
```

```
--- Comparison: Original vs. Stemmed vs. Lemmatized ---
Original Word      | Porter Stem     | NLTK Lemma      | spaCy Lemma
----------------------------------------------------------------------
Diabetes           | diabet          | Diabetes        | diabetes
is                 | is              | is              | be
a                  | a               | a               | a
chronic            | chronic         | chronic         | chronic
disease            | diseas          | disease         | disease
that               | that            | that            | that
affects            | affect          | affect          | affect
how                | how             | how             | how
the                | the             | the             | the
body               | bodi            | body            | body
processes          | process         | process         | process
blood              | blood           | blood           | blood
sugar              | sugar           | sugar           | sugar
.                  | .               | .               | .
----------------------------------------------------------------------
If                 | if              | If              | if
untreated          | untreat         | untreated       | untreate
,                  | ,               | ,               | ,
diabetes           | diabet          | diabetes        | diabete
may                | may             | may             | may
cause              | caus            | cause           | cause
heart              | heart           | heart           | heart
disease            | diseas          | disease         | disease
,                  | ,               | ,               | ,
kidney             | kidney          | kidney          | kidney
failure            | failur          | failure         | failure
,                  | ,               | ,               | ,
nerve              | nerv            | nerve           | nerve
damage             | damag           | damage          | damage
and                | and             | and             | and
vision             | vision          | vision          | vision
problems           | problem         | problem         | problem
.                  | .               | .               | .
----------------------------------------------------------------------
Early              | earli           | Early           | early
diagnosis          | diagnosi        | diagnosis       | diagnosis
and                | and             | and             | and
proper             | proper          | proper          | proper
treatment          | treatment       | treatment       | treatment
help               | help            | help            | help
improve            | improv          | improve         | improve
patient            | patient         | patient         | patient
outcomes           | outcom          | outcome         | outcome
.                  | .               | .               | .
----------------------------------------------------------------------
```