```
!pip install nltk
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.12/dist-packages (3.9.1)
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)
```

```
!pip install spaCy
```

```
Requirement already satisfied: spaCy in /usr/local/lib/python3.12/dist-packages (3.8.11)
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: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.12/dist-packages (from spaCy) (4.67.1)
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: click>=8.0.0 in
/usr/local/lib/python3.12/dist-packages (from typer-
slim<1.0.0,>=0.3.0->spaCy) (8.3.1)
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)
```

```
import nltk
```

```
import spacy



import nltk
nltk.download('punkt')
nltk.download('punkt_tab')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.

True
```

Sentence Tokenization

```
paragraph = "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."
sentences = nltk.sent_tokenize(paragraph)
print("Tokenized sentences:")
for i, sent in enumerate(sentences):
    print(f"{i+1}. {sent}")
```

```
Tokenized sentences:
1. Diabetes is a chronic disease that affects how the body processes
blood sugar.
2. If untreated, diabetes may cause heart disease, kidney failure,
nerve damage and vision problems.
3. Early diagnosis and proper treatment help improve patient outcomes.
```

Word Tokenization

```
import nltk

word_tokenized_sentences = []
print("Word Tokenized Sentences:")
for i, sent in enumerate(sentences):
    words = nltk.word_tokenize(sent)
    word_tokenized_sentences.append(words)
    print(f"Sentence {i+1}: {words}")
```

```
Word Tokenized Sentences:
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', '.']

Stemming

from nltk.stem import PorterStemmer

porter = PorterStemmer()

stemmed_sentences = []
print("Stemmed Sentences:")
for i, words in enumerate(word_tokenized_sentences):
    stemmed_words = [porter.stem(word) for word in words]
    stemmed_sentences.append(stemmed_words)
    print(f"Sentence {i+1}: {stemmed_words}")

Stemmed Sentences:
Sentence 1: ['diabet', 'is', 'a', 'chronic', 'diseas', 'that',
'affect', 'how', 'the', 'bodi', 'process', 'blood', 'sugar', '.']
Sentence 2: ['if', 'untreat', ',', 'diabet', 'may', 'caus', 'heart',
'diseas', ',', 'kidney', 'failur', ',', 'nerv', 'damag', 'and',
'vision', 'problem', '.']
Sentence 3: ['earli', 'diagnosi', 'and', 'proper', 'treatment',
'help', 'improv', 'patient', 'outcom', '.']

import nltk
nltk.download('averaged_perceptron_tagger')
nltk.download('averaged_perceptron_tagger_eng')

[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 averaged_perceptron_tagger_eng to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger_eng.zip.

True
```

Lemmatization

```
from nltk.corpus import wordnet

def get_wordnet_pos(treebank_tag):
    if treebank_tag.startswith('J'):
        return wordnet.ADJ
    elif treebank_tag.startswith('V'):
        return wordnet.VERB
    elif treebank_tag.startswith('N'):
        return wordnet.NOUN
    elif treebank_tag.startswith('R'):
```

```
        return wordnet.ADV
    else:
        return wordnet.NOUN # Default to noun if tag is not found

# Assuming 'paragraph' and 'word_tokenized_sentences' are still in
scope from previous executions.
# If not, they would need to be re-defined/re-run.

lemmatized_sentences_with_pos = []
print("Lemmatized Sentences (with POS tagging):")

for i, words in enumerate(word_tokenized_sentences):
    pos_tags = nltk.pos_tag(words)
    lemmatized_words = []
    for word, tag in pos_tags:
        wntag = get_wordnet_pos(tag)
        lemmatized_words.append(lemmatizer.lemmatize(word, wntag))
    lemmatized_sentences_with_pos.append(lemmatized_words)
    print(f"Sentence {i+1}: {lemmatized_words}")
```

```
Lemmatized Sentences (with POS tagging):
Sentence 1: ['Diabetes', 'be', 'a', 'chronic', 'disease', 'that',
'affect', 'how', 'the', 'body', 'process', 'blood', 'sugar', '.']
Sentence 2: ['If', 'untreated', ',', 'diabetes', 'may', 'cause',
'heart', 'disease', ',', 'kidney', 'failure', ',', 'nerve', 'damage',
'and', 'vision', 'problem', '.']
Sentence 3: ['Early', 'diagnosis', 'and', 'proper', 'treatment',
'help', 'improve', 'patient', 'outcome', '.']
```

Comparision

```
print("\n--- Detailed Word-by-Word Comparison (with POS Lemmatization)
---")
for i in range(len(word_tokenized_sentences)):
    print(f"\nSentence {i+1}:")
    print(f"{'Original':<15} {'Stemmed':<15} {'Lemma (with
POS)':<18}")
    print(f"{'':-<15} {'':-<15} {'':-<18}")

    pos_tags = nltk.pos_tag(word_tokenized_sentences[i])

    for j in range(len(word_tokenized_sentences[i])):
        original_word = word_tokenized_sentences[i][j]
        stemmed_word = stemmed_sentences[i][j]
        # lemmatized_no_pos = lemmatized_sentences[i][j] # This line
is removed

        # Get POS for current word for lemmatization
        wntag = get_wordnet_pos(pos_tags[j][1])
        lemmatized_with_pos = lemmatizer.lemmatize(original_word,
```

```
wntag)

        print(f"{original_word:<15} {stemmed_word:<15}
{lemmatized_with_pos:<18}")

print("\nObservations on POS-aware Lemmatization:")
print("- By providing the correct Part-of-Speech tag, the lemmatizer
can now correctly identify the base form of verbs like 'is' to 'be'.")
print("- This makes lemmatization more linguistically accurate and
useful for tasks requiring precise word forms.")


--- Detailed Word-by-Word Comparison (with POS Lemmatization) ---

Sentence 1:
Original        Stemmed         Lemma (with POS)
--------------- --------------- ------------------
Diabetes        diabet          Diabetes
is              is              be
a               a               a
chronic         chronic         chronic
disease         diseas          disease
that            that            that
affects         affect          affect
how             how             how
the             the             the
body            bodi            body
processes       process         process
blood           blood           blood
sugar           sugar           sugar
.               .               .

Sentence 2:
Original        Stemmed         Lemma (with POS)
--------------- --------------- ------------------
If              if              If
untreated       untreat         untreated
,               ,               ,
diabetes        diabet          diabetes
may             may             may
cause           caus            cause
heart           heart           heart
disease         diseas          disease
,               ,               ,
kidney          kidney          kidney
failure         failur          failure
,               ,               ,
nerve           nerv            nerve
damage          damag           damage
and             and             and
```

```
vision          vision          vision
problems        problem         problem
.               .               .

Sentence 3:
Original        Stemmed         Lemma (with POS)
--------------- --------------- ------------------
Early           earli           Early
diagnosis       diagnosi        diagnosis
and             and             and
proper          proper          proper
treatment       treatment       treatment
help            help            help
improve         improv          improve
patient         patient         patient
outcomes        outcom          outcome
.               .               .

Observations on POS-aware Lemmatization:
- By providing the correct Part-of-Speech tag, the lemmatizer can now
correctly identify the base form of verbs like 'is' to 'be'.
- This makes lemmatization more linguistically accurate and useful for
tasks requiring precise word forms.
```