

```
!pip install nltk spacy
!python -m spacy download en_core_web_sm
```

```
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) (2.10.6)
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)
Requirement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!1.8.1)
Requirement already satisfied: typing-extensions>=4.14.1 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!1.8.1)
Requirement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!1.8.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0)
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)
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)
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)
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)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from Jinja2) (3.0.3)
Requirement already satisfied: wrapt in /usr/local/lib/python3.12/dist-packages (from smart-open<8.0.0,>=5.2.1)
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_core_web_sm-3.8.0-py3-none-any.whl: 12.8/12.8 MB 80.5 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.

```
import nltk
import spacy
from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.stem import PorterStemmer

nltk.download('punkt')
```

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

```
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)
```

```
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(medical_text)
words_nltk = word_tokenize(medical_text)
```

```
print("Sentences (NLTK):", sentences_nltk)
print("\nWords (NLTK):", words_nltk)
```

```
Sentences (NLTK): ['\nDiabetes is a chronic disease that affects how the body processes blood sugar.', 'If untreated, diabete
Words (NLTK): ['Diabetes', 'is', 'a', 'chronic', 'disease', 'that', 'affects', 'how', 'the', 'body', 'processes', 'blood', '']
```

```
doc = nlp(medical_text)
```

```
sentences_spacy = [sent.text for sent in doc.sents]
words_spacy = [token.text for token in doc if not token.is_punct]
```

```
print("Sentences (spaCy):", sentences_spacy)
print("\nWords (spaCy):", words_spacy)
```

```
Sentences (spaCy): ['\nDiabetes is a chronic disease that affects how the body processes blood sugar.\n', 'If untreated, dia
Words (spaCy): ['\n', 'Diabetes', 'is', 'a', 'chronic', 'disease', 'that', 'affects', 'how', 'the', 'body', 'processes', 'bl
```

```
stemmer = PorterStemmer()
```

```
stemmed_words = [stemmer.stem(word) for word in words_nltk if word.isalpha()]
```

```
print("Stemmed Words:")
print(stemmed_words)
```

```
Stemmed Words:
```

```
['diabet', 'is', 'a', 'chronic', 'diseas', 'that', 'affect', 'how', 'the', 'bodi', 'process', 'blood', 'sugar', 'if', 'untre
```

```
lemmatized_words = [token.lemma_ for token in doc if token.is_alpha]
```

```
print("Lemmatized Words:")
print(lemmatized_words)
```

```
Lemmatized Words:
```

```
['Diabetes', 'be', 'a', 'chronic', 'disease', 'that', 'affect', 'how', 'the', 'body', 'process', 'blood', 'sugar', 'if', 'ur
```

```
import nltk
import spacy
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
```

```
nltk.download('punkt')
nlp = spacy.load("en_core_web_sm")
```

```
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.
"""
```

```
tokens = word_tokenize(medical_text)
original_words = [word for word in tokens if word.isalpha()]
```

```
stemmer = PorterStemmer()
stemmed_words = [stemmer.stem(word) for word in original_words]
```

```
doc = nlp(medical_text)
lemmatized_words = [token.lemma_ for token in doc if token.is_alpha]
```

```
print(f"{'ORIGINAL':<15}{'STEMMING':<15}{'LEMMATIZATION'}")
print("-" * 50)
for o, s, l in zip(original_words, stemmed_words, lemmatized_words):
    print(f"{'o':<15}{s:<15}{l}")
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
```

```
[nltk_data] Package punkt is already up-to-date!
```

ORIGINAL	STEMMING	LEMMATIZATION
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
If	if	if
untreated	untreat	untreate
diabetes	diabet	diabete
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
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

```

"""
Aim:
    Compare original words, stemming, and lemmatization for medical text.

Description:
    This code uses NLTK for tokenization and stemming, and spaCy for lemmatization.
    It demonstrates why lemmatization is better for medical/healthcare NLP.
"""

import nltk
import spacy
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer

# Download required NLTK data
nltk.download('punkt')

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

# Medical text
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.
"""

# -----
# Step 1: Tokenization
# -----
tokens = word_tokenize(medical_text) # Split text into words
original_words = [word for word in tokens if word.isalpha()] # Remove punctuation

# -----
# Step 2: Stemming using NLTK
# -----
stemmer = PorterStemmer()
stemmed_words = [stemmer.stem(word) for word in original_words]

# -----
# Step 3: Lemmatization using spaCy
# -----
doc = nlp(medical_text)
lemmatized_words = [token.lemma_ for token in doc if token.is_alpha]

# -----
# Step 4: Display results
# -----
print(f"{'ORIGINAL':<15}{'STEMMING':<15}{'LEMMATIZATION'}")
print("-" * 50)

for o, s, l in zip(original_words, stemmed_words, lemmatized_words):
    print(f"{o:<15}{s:<15}{l}")

```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
```

```
[nltk_data] Package punkt is already up-to-date!
```

ORIGINAL	STEMMING	LEMMATIZATION
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
If	if	if
untreated	untreat	untreate
diabetes	diabet	diabete
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
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

```
"""
```

Medical Text Preprocessing using Tokenization, Stemming, and Lemmatization.

This script demonstrates preprocessing of medical or healthcare-related text.

It performs the following steps:

1. Tokenization - splitting text into words
2. Stemming - reducing words to their root form (NLTK)
3. Lemmatization - converting words to their meaningful base/dictionary form (spaCy)
4. Compares outputs to show why lemmatization is preferred in healthcare NLP

Requirements:

- Python 3.x
- Libraries: nltk, spacy
- spaCy model: en\_core\_web\_sm

Author: [Your Name]

Date: [Today's Date]

```
"""
```

```
import nltk
import spacy
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
```

```
def preprocess_medical_text(text):
```

```
    """
```

Preprocess a given medical text and compare original, stemmed, and lemmatized words.

Args:

text (str): A string containing medical or healthcare-related text.

Returns:

None. Prints a table comparing original words, stemmed words, and lemmatized words.

Steps:

1. Tokenizes text into words (ignoring punctuation).
2. Applies stemming using NLTK's PorterStemmer.
3. Applies lemmatization using spaCy.
4. Prints results in a formatted table.

```
    """
```

```
    # Download NLTK data for tokenization (run once)
    nltk.download('punkt', quiet=True)
```

```
    # Load spaCy English model
```

```

nlp = spacy.load("en_core_web_sm")

# -----
# Tokenization
# -----
tokens = word_tokenize(text)
original_words = [word for word in tokens if word.isalpha()]

# -----
# Stemming (NLTK)
# -----
stemmer = PorterStemmer()
stemmed_words = [stemmer.stem(word) for word in original_words]

# -----
# Lemmatization (spaCy)
# -----
doc = nlp(text)
lemmatized_words = [token.lemma_ for token in doc if token.is_alpha]

# -----
# Display Results
# -----
print(f"{'ORIGINAL':<15}{'STEMMING':<15}{'LEMMATIZATION':<15}")
print("-" * 50)
for o, s, l in zip(original_words, stemmed_words, lemmatized_words):
    print(f"{o:<15}{s:<15}{l}")

# -----
# Example Usage
# -----
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.
"""

preprocess_medical_text(medical_text)

```

ORIGINAL	STEMMING	LEMMATIZATION
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
If	if	if
untreated	untreat	untreated
diabetes	diabet	diabete
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
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

