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

```
!pip install nltk spacy
```

```
rement already satisfied: nltk in /usr/local/lib/python3.12/dist-packages (3.9.1)
rement already satisfied: spacy in /usr/local/lib/python3.12/dist-packages (3.8.11)
rement already satisfied: click in /usr/local/lib/python3.12/dist-packages (from nltk) (8.3.1)
rement already satisfied: joblib in /usr/local/lib/python3.12/dist-packages (from nltk) (1.5.3)
rement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.12/dist-packages (from nltk) (2
rement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from nltk) (4.67.1)
rement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.12/dist-packages (f
rement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.12/dist-packages (f
rement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.12/dist-packages (fro
rement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.12/dist-packages (from spac
rement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.12/dist-packages (from sp
rement already satisfied: thinc<8.4.0,>=8.3.4 in /usr/local/lib/python3.12/dist-packages (from spac
rement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.12/dist-packages (from spa
rement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.12/dist-packages (from spac
rement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.12/dist-packages (from
rement already satisfied: weasel<0.5.0,>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from spa
rement already satisfied: typer-slim<1.0.0,>=0.3.0 in /usr/local/lib/python3.12/dist-packages (from
rement already satisfied: numpy>=1.19.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.
rement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.12/dist-packages (from
rement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.12/dist-pa
rement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.1.6)
rement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from spacy) (75.2.
rement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (
rement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from p
rement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (from py
rement already satisfied: typing-extensions>=4.14.1 in /usr/local/lib/python3.12/dist-packages (fro
rement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from
rement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from
rement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests<3.
rement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from reque
rement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from reque
rement already satisfied: blis<1.4.0,>=1.3.0 in /usr/local/lib/python3.12/dist-packages (from thinc
rement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.12/dist-packages (from
rement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.12/dist-packages (fr
rement already satisfied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.12/dist-packages (from
rement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->
rement already satisfied: wrapt in /usr/local/lib/python3.12/dist-packages (from smart-open<8.0.0,>
```

```
import nltk
nltk.download('punkt')
nltk.download('punkt_tab') # Added to resolve LookupError
import spacy
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
```

```
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.
"""
```

Comparison of Stemming and Lemmatization Outputs

Original Word (from NLTK)	NLTK Stemmed (PorterStemmer)	spaCy Lemmatized (simplified)
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
.	.	.

Observations from this comparison:

- **NLTK Stemming (Porter Stemmer):** Tends to be more aggressive, often chopping off suffixes to reach a root form that might not be a valid word (e.g., 'Diabetes' to 'diabet', 'disease' to 'diseas', 'processes' to 'process', 'body' to 'bodi', 'untreated' to 'untreat', 'outcomes' to 'outcom'). It's a rule-based approach.
- **spaCy Lemmatization:** Generally produces a valid base form (lemma) of the word, often looking it up in a dictionary-like structure (e.g., 'Diabetes' remains 'Diabetes', 'is' becomes 'be', 'processes' becomes 'process', 'body' remains 'body'). It's more sophisticated and context-aware.

Why Lemmatization is Critical in Healthcare NLP:

1. **Meaning Preservation:** In healthcare, precision is paramount. Lemmatization ensures that words are reduced to their meaningful base form, which is crucial for retaining clinical context. Aggressive stemming can strip away too much, turning 'infection' into 'infect' or 'infectious' into 'infect', which might lose subtle but important distinctions or even result in non-words that hinder interpretation. For example, 'diabet' from 'diabetes' is not a common medical term, whereas 'diabetes' as a lemma retains its full clinical meaning.
2. **Accuracy in Clinical Information Extraction:** When extracting conditions, symptoms, treatments, or medications from clinical notes, having the correct base form of a word is essential. A system searching for 'diagnosed' or 'diagnosing' should correctly identify 'diagnosis' as the core concept. Lemmatization handles this effectively, reducing all inflected forms to 'diagnosis', thereby improving the accuracy of information retrieval and entity recognition.
3. **Integration with Medical Terminologies and Ontologies:** Healthcare relies heavily on standardized vocabularies like SNOMED CT, ICD codes, and UMLS. These terminologies are built around precise medical concepts. Lemmatization helps in mapping natural language terms from patient notes to these standardized concepts, ensuring that variations like 'kidney failures' and 'kidney failure' are correctly linked to the same underlying medical condition. Stemming's non-word outputs would make this mapping much more challenging and error-prone.
4. **Clinical Decision Support Systems (CDSS):** CDSS often analyze patient data to provide recommendations or alerts. Misinterpretations arising from stemming (e.g., 'complications' becoming 'complic') could lead to incorrect medical advice, missed diagnoses, or inappropriate treatment suggestions, potentially harming patients. Lemmatization provides a more reliable foundation for such critical systems.
5. **Data Quality and Analytics:** For downstream analysis, machine learning models, or statistical reporting, consistent representation of terms is vital. Lemmatization provides a higher quality, more consistent input for these processes, leading to more robust and accurate insights from healthcare data. It reduces noise without sacrificing meaning.

In essence, while stemming is a faster, simpler approach, its heuristic nature can be too imprecise for the high-stakes environment of healthcare. Lemmatization's ability to maintain semantic integrity by reducing words to their dictionary forms makes it a superior and often critical choice for reliable Natural Language Processing in medical and clinical applications.

Stemming with NLTK

```
from nltk.stem import PorterStemmer
```

```
stemmer = PorterStemmer()
nltk_stemmed_words = [stemmer.stem(word) for word in nltk_words]

print("Original NLTK Words:", nltk_words)
print("NLTK Stemmed Words:", nltk_stemmed_words)
```

```
Original NLTK Words: ['Diabetes', 'is', 'a', 'chronic', 'disease', 'that', 'affects', 'how', 'the'
NLTK Stemmed Words: ['diabet', 'is', 'a', 'chronic', 'diseas', 'that', 'affect', 'how', 'the', 'bc
```

Lemmatization with spaCy

```
# Assuming nlp and spacy_doc are already loaded and updated from previous steps.
# If not, please re-run the spaCy tokenization cell first.

spacy_lemmas = [token.lemma_ for token in spacy_doc]

print("Original spaCy Words:", spacy_words)
print("spaCy Lemmatized Words:", spacy_lemmas)
```

```
Original spaCy Words: ['\n', 'Diabetes', 'is', 'a', 'chronic', 'disease', 'that', 'affects', 'how'
spaCy Lemmatized Words: ['\n', 'Diabetes', 'be', 'a', 'chronic', 'disease', 'that', 'affect', 'how
```

Stemming with NLTK

```
from nltk.stem import PorterStemmer

stemmer = PorterStemmer()
nltk_stemmed_words = [stemmer.stem(word) for word in nltk_words]

print("Original NLTK Words:", nltk_words)
print("NLTK Stemmed Words:", nltk_stemmed_words)
```

```
Original NLTK Words: ['Diabetes', 'is', 'a', 'chronic', 'disease', 'that', 'affects', 'how', 'the'
NLTK Stemmed Words: ['diabet', 'is', 'a', 'chronic', 'diseas', 'that', 'affect', 'how', 'the', 'bc
```

Lemmatization with spaCy

```
# Assuming nlp and spacy_doc are already loaded and updated from previous steps
# If they haven't been updated for the new medical_text, please re-run the spaCy tokenization cel

spacy_lemmas = [token.lemma_ for token in spacy_doc]

print("Original spaCy Words:", spacy_words)
print("spaCy Lemmatized Words:", spacy_lemmas)
```

```
Original spaCy Words: ['\n', 'Diabetes', 'is', 'a', 'chronic', 'disease', 'that', 'affects', 'how'
spaCy Lemmatized Words: ['\n', 'Diabetes', 'be', 'a', 'chronic', 'disease', 'that', 'affect', 'how
```

NLTK Tokenization

```
nltk_sentences = nltk.sent_tokenize(medical_text)
nltk_words = nltk.word_tokenize(medical_text)
```

```
print("NLTK Sentence Tokenization:", nltk_sentences)
print("NLTK Word Tokenization:", nltk_words)
```

```
NLTK Sentence Tokenization: ['\nDiabetes is a chronic disease that affects how the body processes
NLTK Word Tokenization: ['Diabetes', 'is', 'a', 'chronic', 'disease', 'that', 'affects', 'how', 't
```

spaCy Tokenization

```
# Load the small English model. Download it if not already present.
try:
    nlp = spacy.load('en_core_web_sm')
except OSError:
    print('Downloading spaCy model en_core_web_sm...')
    spacy.cli.download('en_core_web_sm')
    nlp = spacy.load('en_core_web_sm')

spacy_doc = nlp(medical_text)

spacy_sentences = [sent.text for sent in spacy_doc.sents]
spacy_words = [token.text for token in spacy_doc]

print("spaCy Sentence Tokenization:", spacy_sentences)
print("spaCy Word Tokenization:", spacy_words)
```

```
spaCy Sentence Tokenization: ['\nDiabetes is a chronic disease that affects how the body processes
spaCy Word Tokenization: ['\n', 'Diabetes', 'is', 'a', 'chronic', 'disease', 'that', 'affects', 'h
```