

Text Processing

Name		Date	
Class / Sec		Roll no.	

1 • AIM

To write a Python program that performs **text normalisation** on a piece of text — splitting it into sentences and tokens, removing stop words, converting to lowercase, and reducing words to their base form.

2 • SOFTWARE AND LIBRARIES USED

TOOL	WHAT IT IS FOR
Python 3	The programming language the program is written in
Jupyter Notebook	The environment used to write and run the code (IDLE or Colab also acceptable)
re (built in)	Python's regular expression module, used here to split sentences and tokens

3 • THEORY

The language of computers is **numerical**. Before text can become numbers it has to be simplified, and that process is called **Text Normalisation**. It reduces the data to a level where its complexity is lower than the original.

#	STEP	WHAT IT DOES
1	Sentence Segmentation	Divides the whole corpus into sentences
2	Tokenization	Divides each sentence into tokens — words, numbers, special characters
3	Removing stop words	Drops frequent words that carry no meaning, plus punctuation
4	Converting to a common case	Usually lowercase, so Hello and hello are the same token
5	Stemming or Lemmatization	Reduces words to a base form. These two are alternatives , not consecutive steps

Stemming or lemmatization — not both: **Stemming** just strips affixes, so “studies” becomes **studi**, which is not a word. **Lemmatization** guarantees a real word, giving **study**, but takes longer to run. You choose one.

4 • PROGRAM

```
import re

text = "Aditi and Riya are studying. Riya studied all night! Aditi is a good student."
print("Original text :", text)
print("Word count      :", len(text.split()))

# Step 1 - Sentence segmentation
sentences = re.split(r'(?=[.!?])\s+', text)
print("\n1. Sentences:", sentences)

# Step 2 - Tokenization
tokens = re.findall(r'\w+|[\^w\s]', text)
print("\n2. Tokens:", tokens)
```

```

print("    Total tokens:", len(tokens))

# Step 3 - Remove stop words and punctuation
stopwords = ["and", "are", "is", "a", "all", "the", "to"]
kept = [t for t in tokens if t.isalpha() and t.lower() not in stopwords]
print("\n3. After removing stop words:", kept)

# Step 4 - Convert to lowercase
lower = [t.lower() for t in kept]
print("\n4. Lowercase:", lower)

# Step 5 - Lemmatization (simple dictionary version)
lemmas = {"studying": "study", "studied": "study"}
final = [lemmas.get(w, w) for w in lower]
print("\n5. After lemmatization:", final)
print("    Unique tokens:", sorted(set(final)))

```

Why a dictionary for step 5: Real lemmatization uses a library such as **NLTK** or **spaCy**. A small dictionary is used here so the step is visible and the program runs anywhere without extra installation.

5 • EXPECTED OUTPUT

The program prints the text after each of the five steps.

```

Original text : Aditi and Riya are studying. Riya studied all night! Aditi is a good student.
Word count    : 14

1. Sentences: ['Aditi and Riya are studying.', 'Riya studied all night!', 'Aditi is a good student.']

2. Tokens: ['Aditi', 'and', 'Riya', 'are', 'studying', '.', 'Riya', 'studied', 'all', 'night', '!', 'Aditi', 'is', 'a', 'good', 'student', '.']
Total tokens: 17

3. After removing stop words:
['Aditi', 'Riya', 'studying', 'Riya', 'studied', 'night', 'Aditi', 'good', 'student']

4. Lowercase:
['aditi', 'riya', 'studying', 'riya', 'studied', 'night', 'aditi', 'good', 'student']

5. After lemmatization:
['aditi', 'riya', 'study', 'riya', 'study', 'night', 'aditi', 'good', 'student']
Unique tokens: ['aditi', 'good', 'night', 'riya', 'student', 'study']

```

[PASTE SCREENSHOT HERE]

Paste a screenshot of YOUR program running, with the code and output visible together

6 • FOLLOWING THE TEXT THROUGH

Watch the count fall at every stage. This is the whole point of normalisation.

AFTER THIS STEP	COUNT	WHAT CHANGED
Original text	14 words	Nothing yet — this is the starting point
1 • Segmentation	3 sentences	Split at the full stops and the exclamation mark
2 • Tokenization	17 tokens	More than 14 — because punctuation counts as tokens

		too
3 • Stop words removed	9 tokens	Gone: and, are, all, is, a, and the three punctuation marks
4 • Lowercase	9 tokens	Same count — only the letters changed
5 • Lemmatization	6 unique	studying and studied both became study

Notice step 2: The count went **up**, from 14 words to 17 tokens. That is correct — the three punctuation marks are tokens. It is exactly why step 3 has to remove them.

7 • OBSERVATION AND RESULT

Two or three sentences in your own words. Say what happened to the text, and what was lost along the way.

Example observation: The original 14-word text produced 17 tokens, because punctuation is counted as tokens too. Removing stop words and punctuation cut this to 9, and lemmatization reduced studying and studied to the single form study, leaving 6 unique tokens. The text is now much simpler, although grammar and word order have been lost.

Worth knowing: Normalisation **throws information away on purpose**. Grammar, word order and tense are all gone — we no longer know that Riya was the one studying. What survives is the **subject matter**, which is enough for tasks like classification.

8 · VIVA QUESTIONS FOR THIS PROGRAM

LIKELY QUESTION	SHAPE OF A STRONG ANSWER
Why are there 17 tokens from 14 words?	Punctuation counts as tokens. Two full stops and one exclamation mark make three extra
What are stop words, and why remove them?	Words that occur very frequently but add no value — and, is, are. Removing them leaves the meaningful terms
Why convert everything to lowercase?	So the machine does not treat Hello and hello as two different words
Difference between stemming and lemmatization?	Both remove affixes. Lemmatization guarantees a real word but is slower. “studies” gives studi or study
What has been lost by the end?	Grammar, word order and tense. We know which words appeared, not who did what

9 · HOW THIS ENTRY IS MARKED

FULL MARKS

All five steps shown separately with output after each · commented code · a screenshot of **your own** output · an observation tracking the **falling token count** · you can say what was lost

LOSES MARKS

Only the final list printed, with no intermediate steps · no screenshot, or someone else's · an observation that just repeats the token list · thinks stemming and lemmatization are done one after the other

Before you submit: Use **your own** sentences — three or four lines about anything. Make sure at least one word appears **twice in different forms** (study/studying, play/played), otherwise step 5 does nothing visible and you cannot explain it in the viva.