Introduction to natural language processing

NATURAL LANGUAGE PROCESSING (NLP) IN PYTHON

Fouad Trad
Machine Learning Engineer

datacamp

Meet the instructor...

Fouad Trad

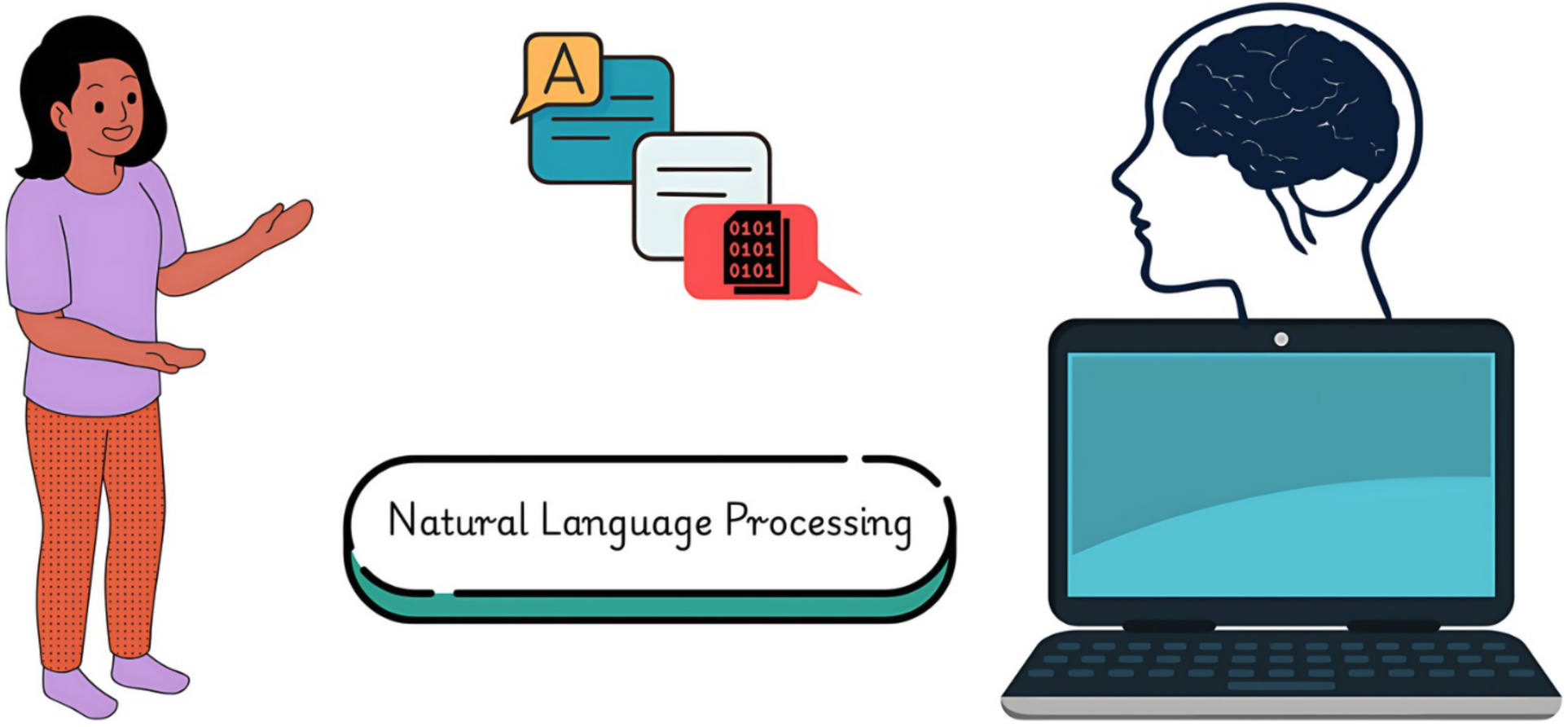
- Machine learning engineer
- Research scientist
- NLP in cybersecurity and healthcare

What is NLP?

- Language is our primary way to communicate
- Computers don't understand our language

What is NLP?

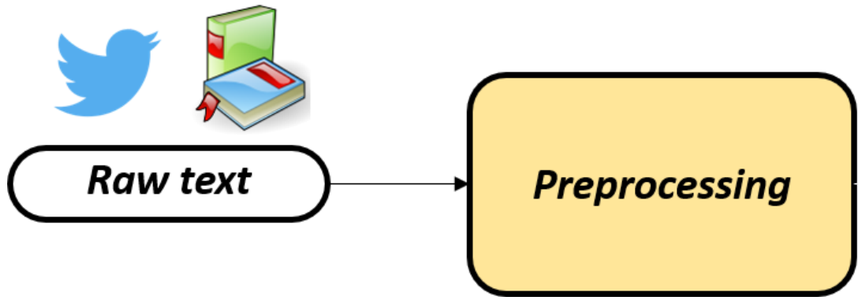
Enables computers to analyze human language



NLP workflow

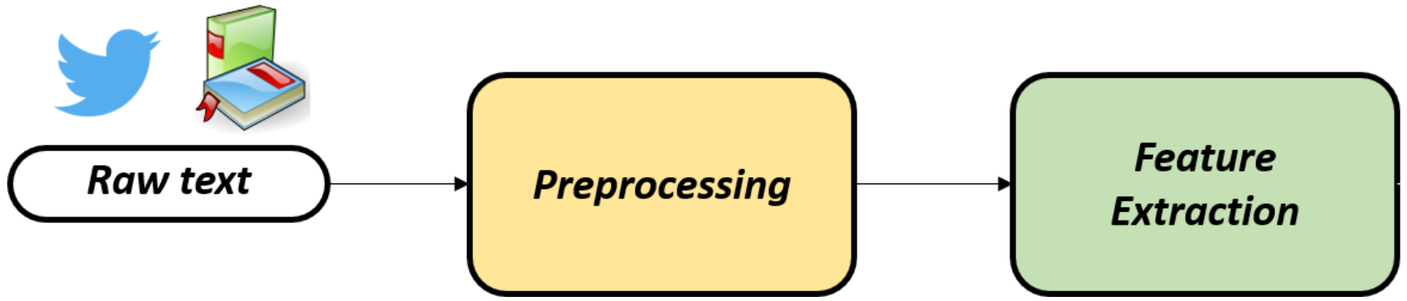
- **Raw text:** anything from tweet to book paragraph

NLP workflow



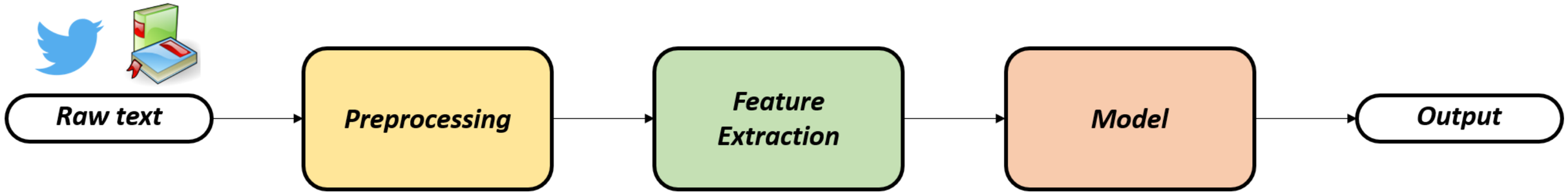
- **Raw text:** anything from tweet to book paragraph
- **Preprocessing:** cleaning text and removing unnecessary elements

NLP workflow



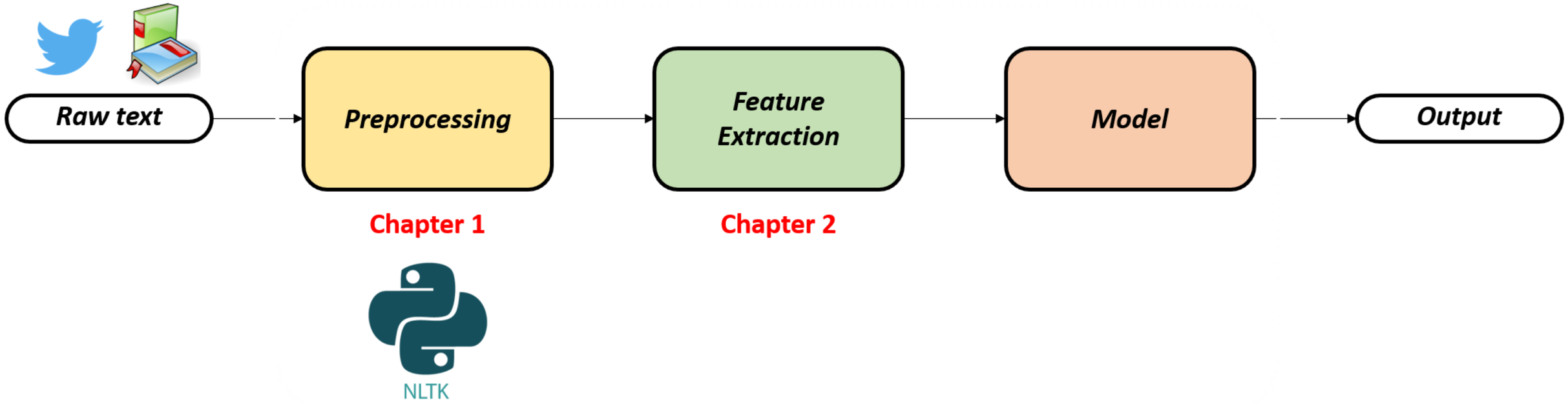
- **Raw text:** anything from tweet to book paragraph
- **Preprocessing:** cleaning text and removing unnecessary elements
- **Feature extraction:** converting text into numbers

NLP workflow

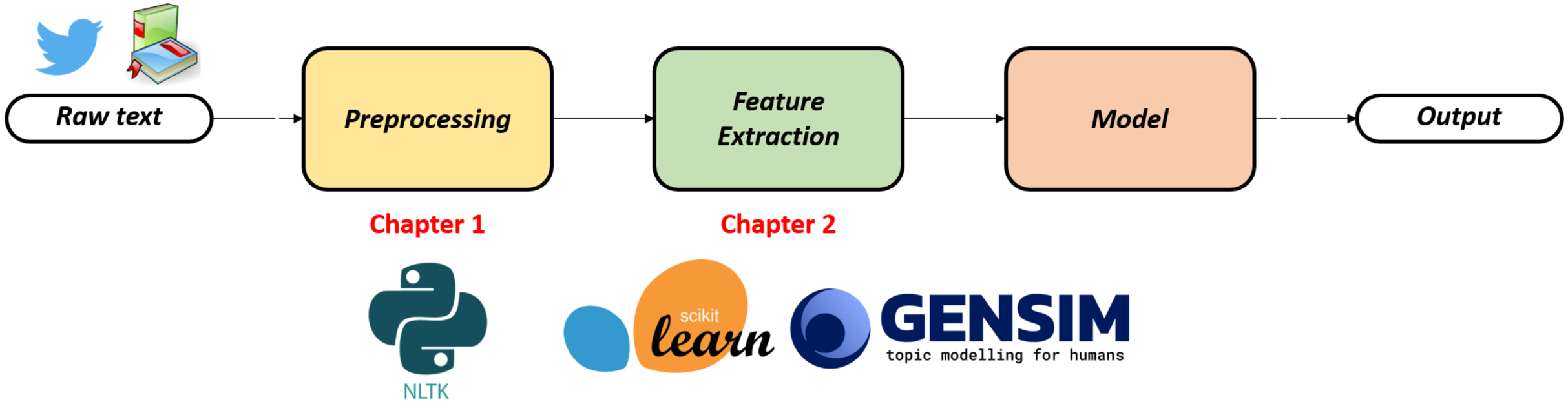


- **Raw text:** anything from tweet to book paragraph
- **Preprocessing:** cleaning text and removing unnecessary elements
- **Feature extraction:** converting text into numbers
- **Model:** analyze, predict, classify, generate new content

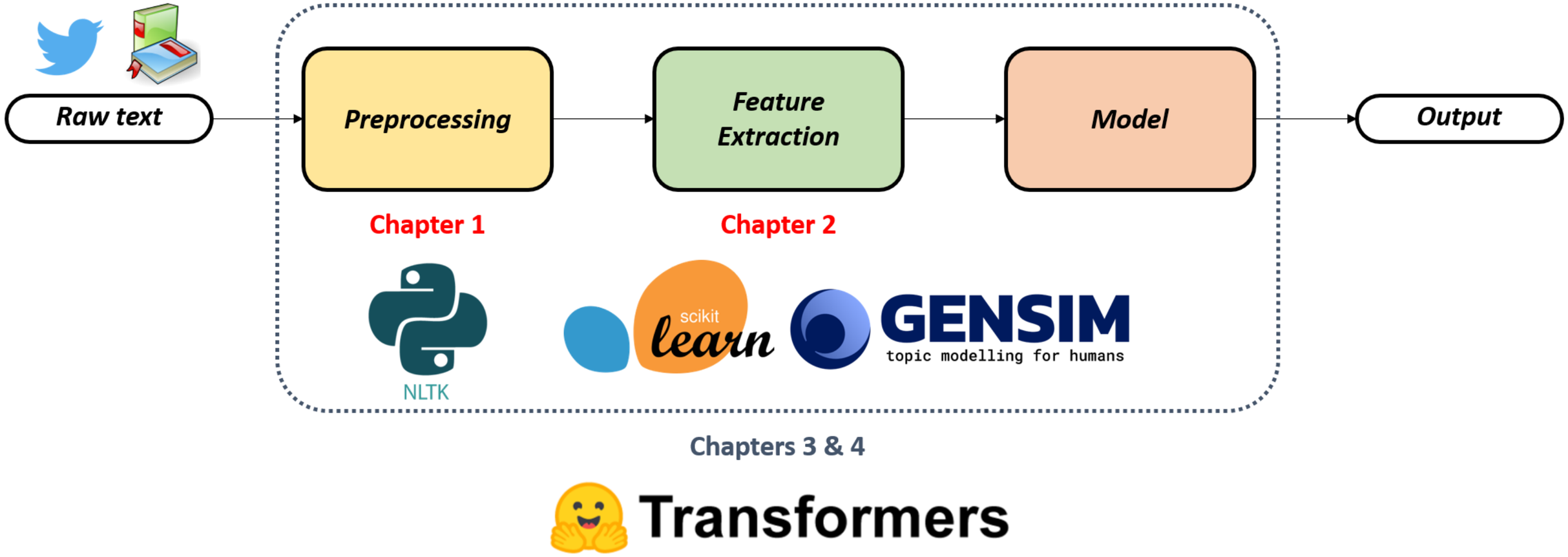
Course plan


Raw text
Preprocessing
Feature Extraction
Model
Output
Chapter 1
Chapter 2
NLTK

Course plan


Raw text
Preprocessing
Feature Extraction
Model
Output
Chapter 1
Chapter 2
NLTK
scikit learn
GENSIM
topic modelling for humans

Course plan


Raw text
Preprocessing
Feature Extraction
Model
Output
Chapter 1
Chapter 2
NLTK
scikit learn
GENSIM
topic modelling for humans
Chapters 3 & 4
Transformers

Tokenization

- Breaks text into tokens (smaller manageable pieces)

Sentence tokenization

- Text → sentences
- Offers clearer insights than analyzing the text as a whole

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

text = "NLP is fun. Let's dive into it!"
sentences = nltk.sent_tokenize(text)
print(sentences)
```

```
["NLP is fun.", "Let's dive into it!"]
```

Word tokenization

- Text → words and punctuation
- Useful for tasks that require:
  - Identifying key terms
  - Counting word frequency

```
text = "Claim your free prize now!"
words = nltk.word_tokenize(text)
print(words)
```

```
['Claim', 'your', 'free', 'prize', 'now', '!']
```

Let's practice!

NATURAL LANGUAGE PROCESSING (NLP) IN PYTHON

Stop words and punctuation handling

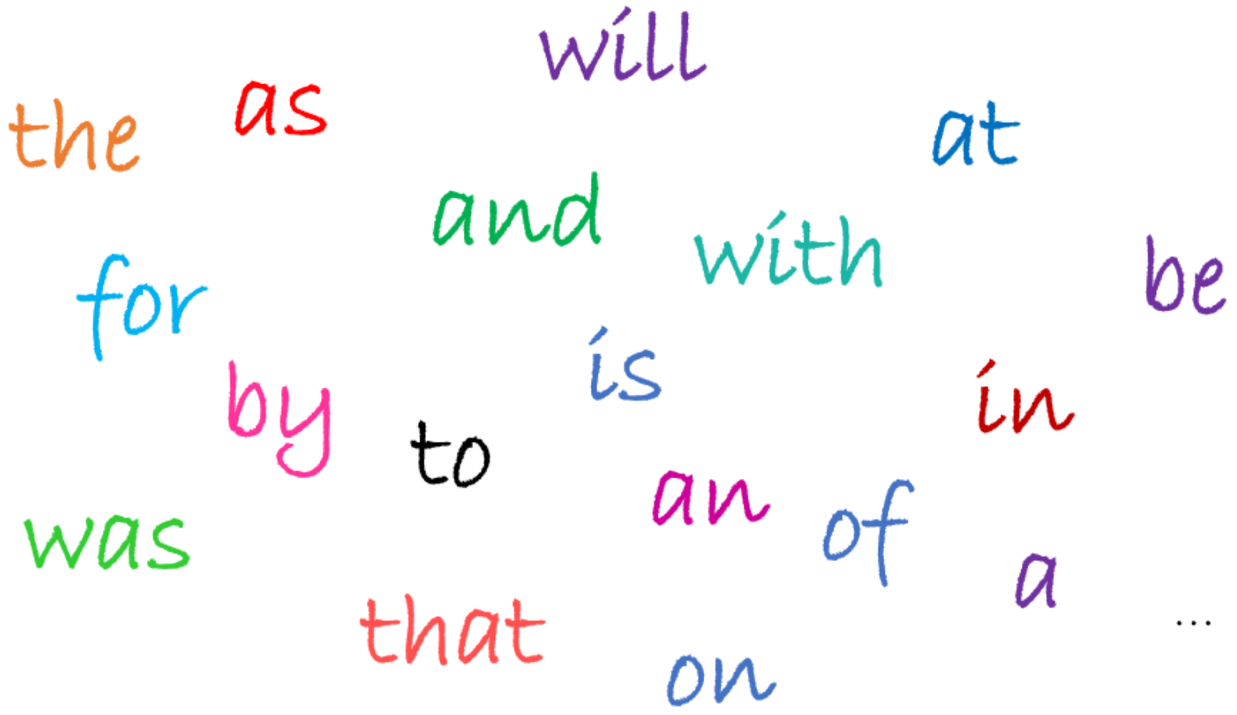
NATURAL LANGUAGE PROCESSING (NLP) IN PYTHON

Fouad Trad
Machine Learning Engineer

datacamp

Stop words

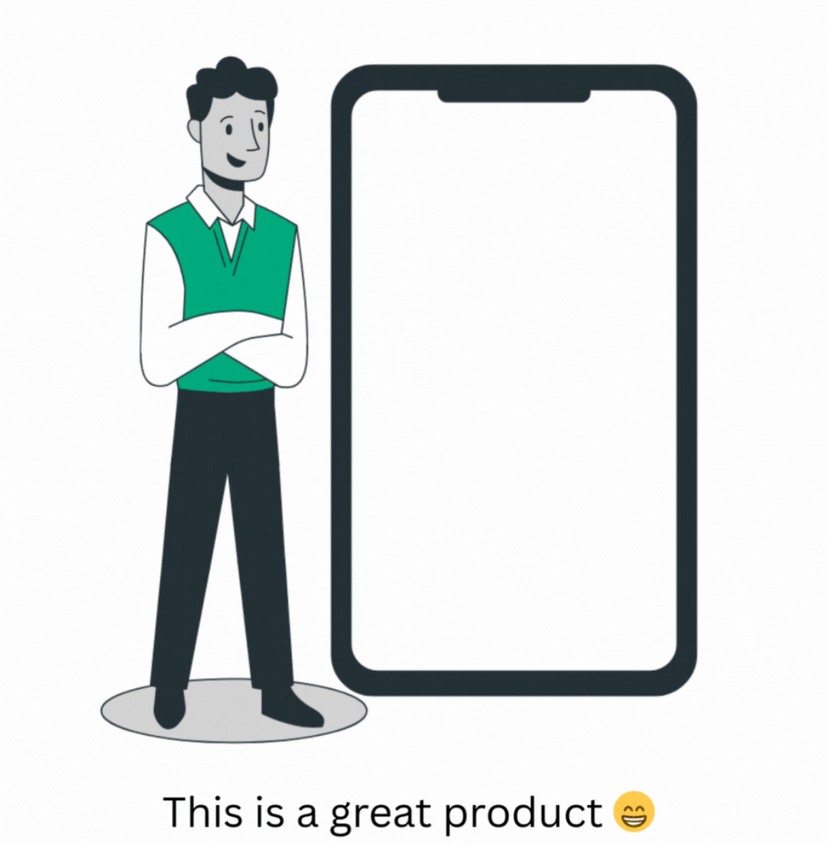
- Appear frequently but contribute little to the machine's understanding of context
- Don't add much value in many NLP tasks
- Removing them helps models focus on important words



Stop words removal

Useful for

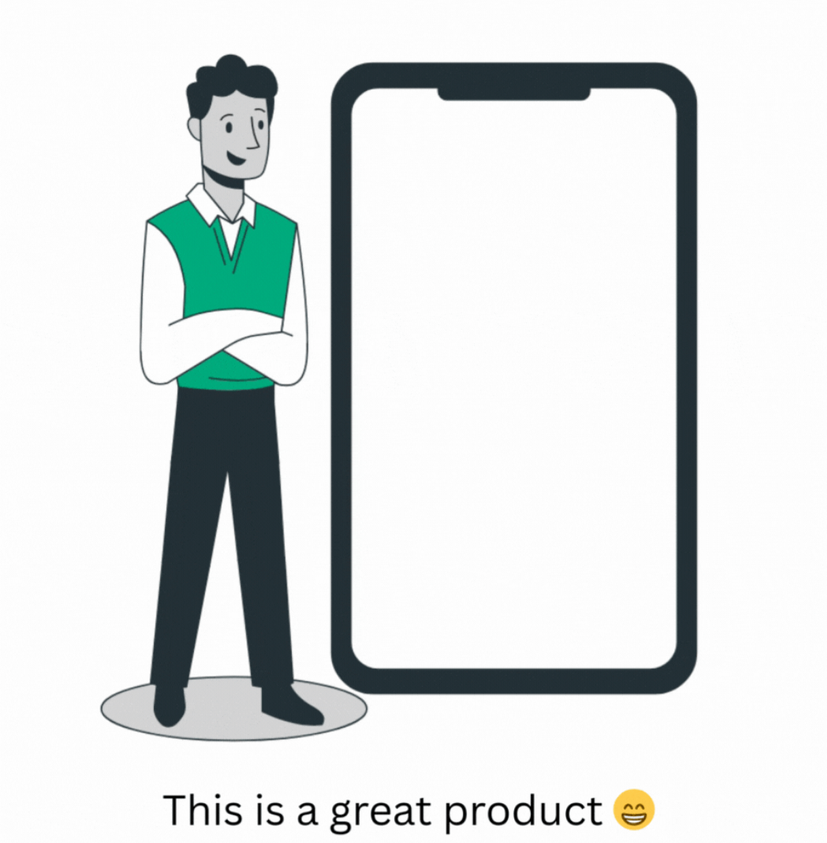
Understanding the topic of a text



Stop words removal

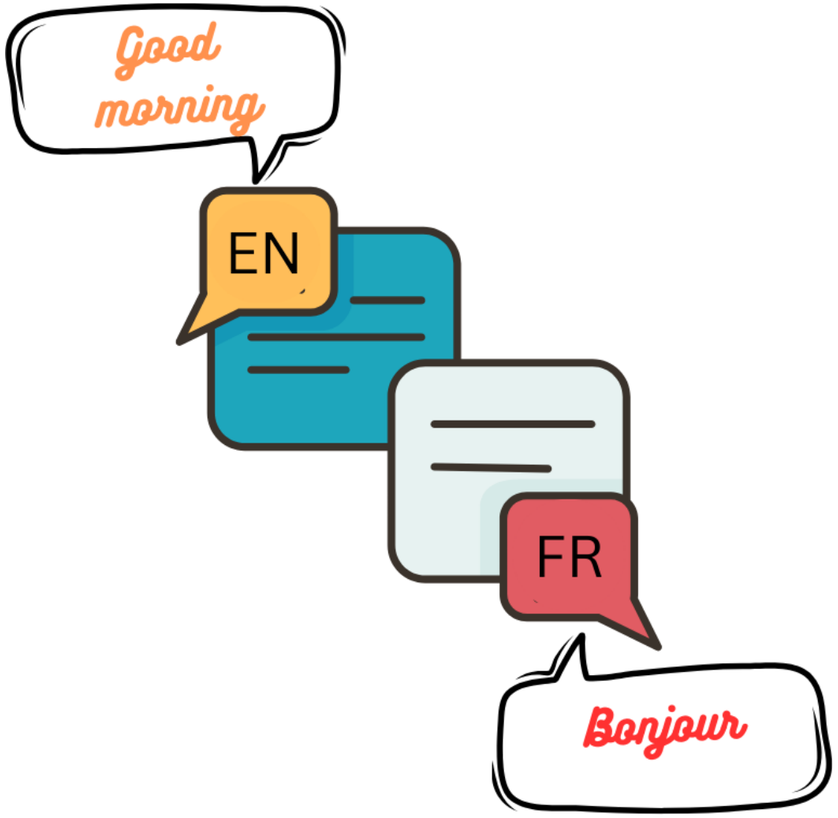
Useful for

Understanding the topic of a text



Not useful for

Tasks requiring every word in the text



Accessing stop words

NLTK provides a list of stop words for several languages

```
from nltk.corpus import stopwords
nltk.download('stopwords')

stop_words = stopwords.words('english')
print(stop_words[:10])
```

```
['a', 'about', 'above', 'after', 'again', 'against', 'ain', 'all', 'am', 'an']
```

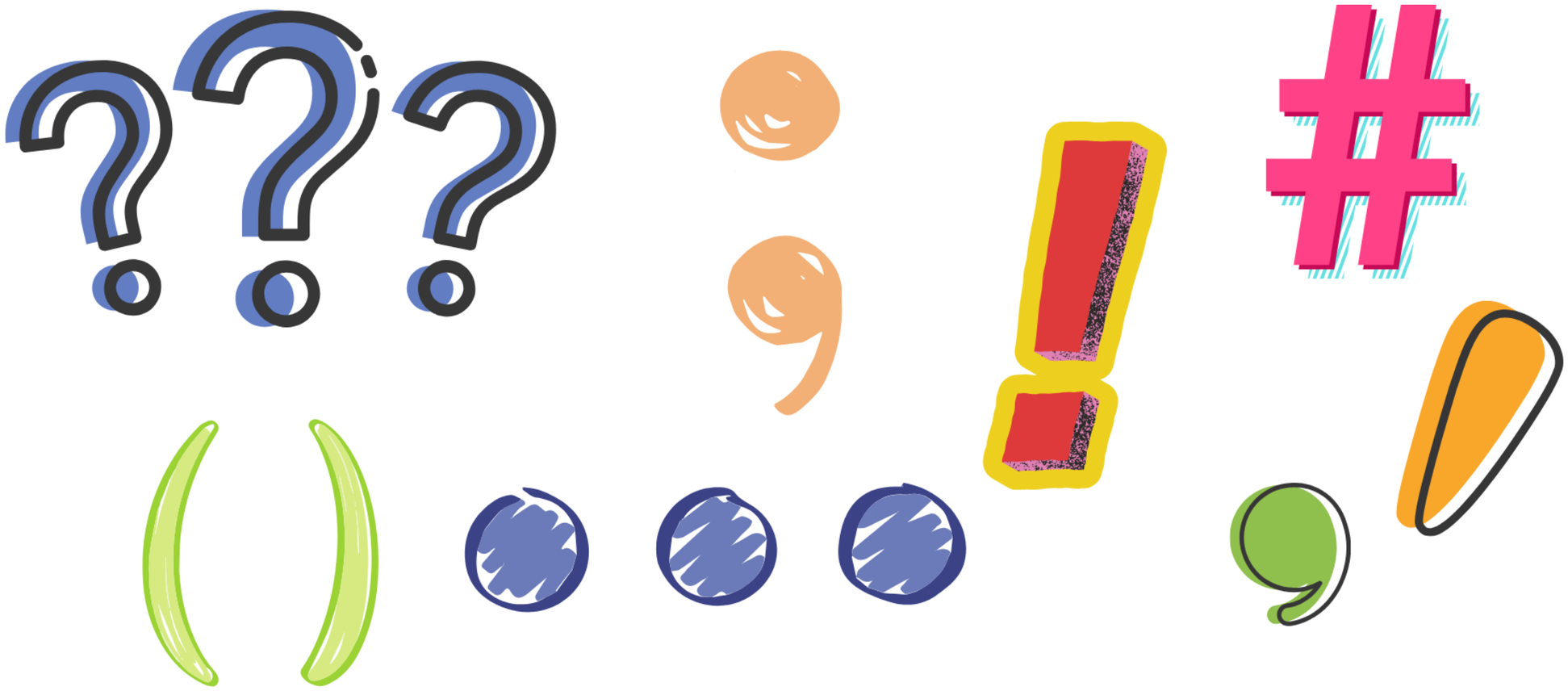
Removing stop words

```
from nltk.tokenize import word_tokenize


text = "This is an example to demonstrate removing stop words."
tokens = word_tokenize(text)
# The .lower() method helps with case sensitivity
filtered_tokens = [word for word in tokens if word.lower() not in stop_words]
print(filtered_tokens)
```

```
['example', 'demonstrate', 'removing', 'stop', 'words', '.']
```

Punctuation

- Structuring language for humans
- No meaningful information in many NLP tasks



Punctuation removal

Useful for

Tasks requiring to find common or important words in documents

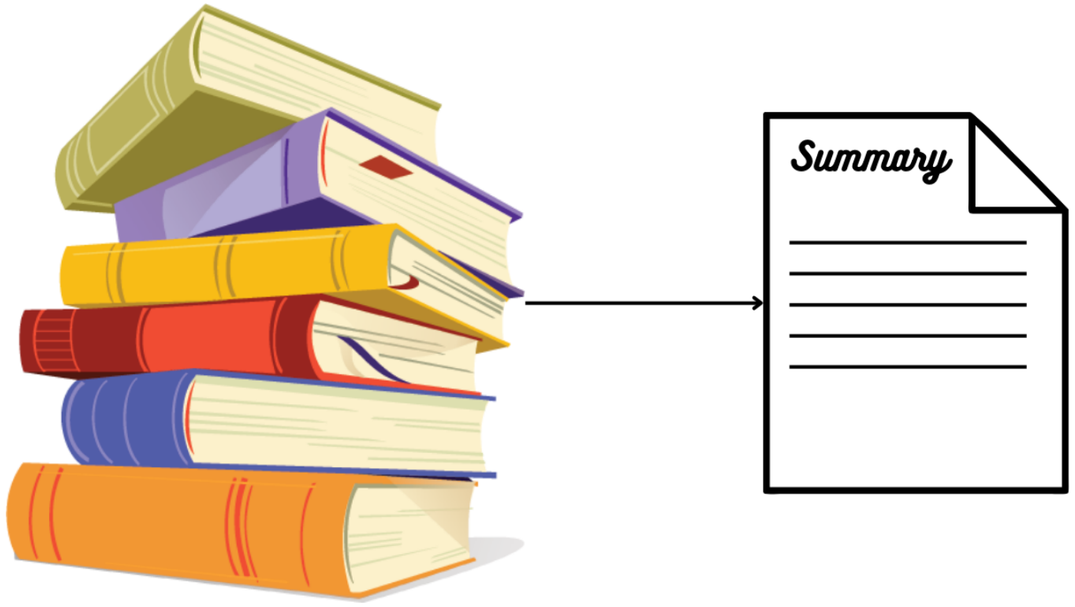
Punctuation removal

Useful for

Tasks requiring to find common or important words in documents

Not useful for

Tasks requiring to maintain sentence structure for clarity



Accessing and removing punctuation

```
import string
print(string.punctuation)
```

```
!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
```

```
text = "This is an example to demonstrate removing stop words."
tokens = word_tokenize(text)
filtered_tokens = [word for word in tokens if word.lower() not in stop_words]

clean_tokens = [word for word in filtered_tokens if word not in string.punctuation]
print(clean_tokens)
```

```
['example', 'demonstrate', 'removing', 'stop', 'words']
```

Let's practice!

NATURAL LANGUAGE PROCESSING (NLP) IN PYTHON

Text normalization techniques

NATURAL LANGUAGE PROCESSING (NLP) IN PYTHON

Fouad Trad
Machine Learning Engineer

datacamp

Text normalization

- Normalization brings words into standard format
- Useful in text classification, search, sentiment analysis
- Techniques to explore:
  - Lowercasing
  - Stemming
  - Lemmatization

I enjoy watching movies with great direction!

I enjoyed watching movies that are well-directed.

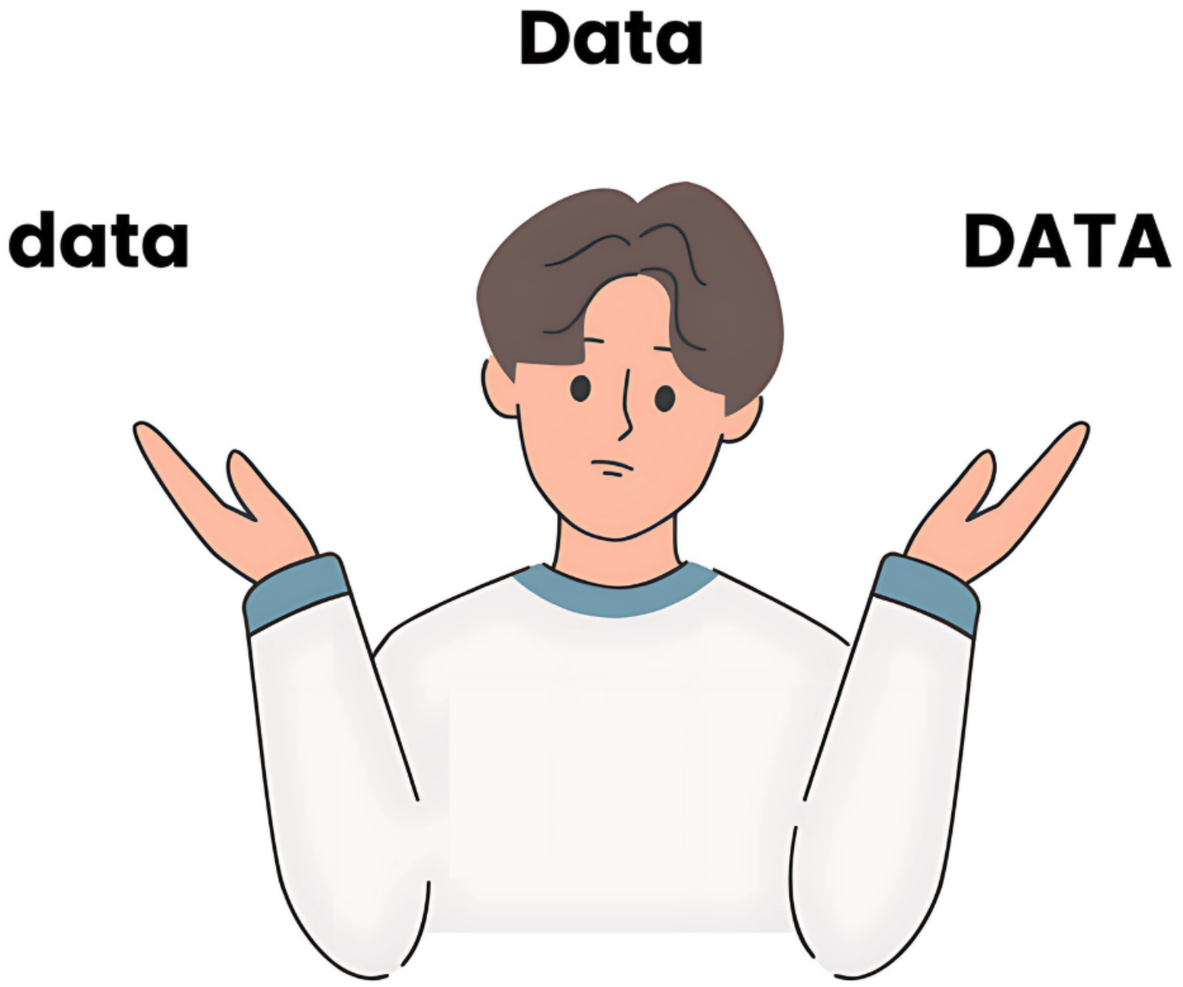
Movies with excellent direction are something I enjoy.

Lowercasing

- If we don't normalize, these words will be treated as separate tokens
- This will mislead models and inflate vocabulary size

```
text  = """Data scientists and data
engineers need DATA"""
lower_text = text.lower()
print(lower_text)
```

```
data scientists and data
engineers need data
```



Lowercasing applicability

Useful for

Tasks relying on keyword analysis

Lowercasing applicability

Useful for

Tasks relying on keyword analysis

Not useful for

Tasks requiring distinction between words that depend on case for meaning

us

US

Stemming

- Reduces words to their root form by chopping off suffixes
  - running → run
  - reading → read
- Useful for grouping similar variations of word together
- Can result in non-valid words
  - organization → organizat

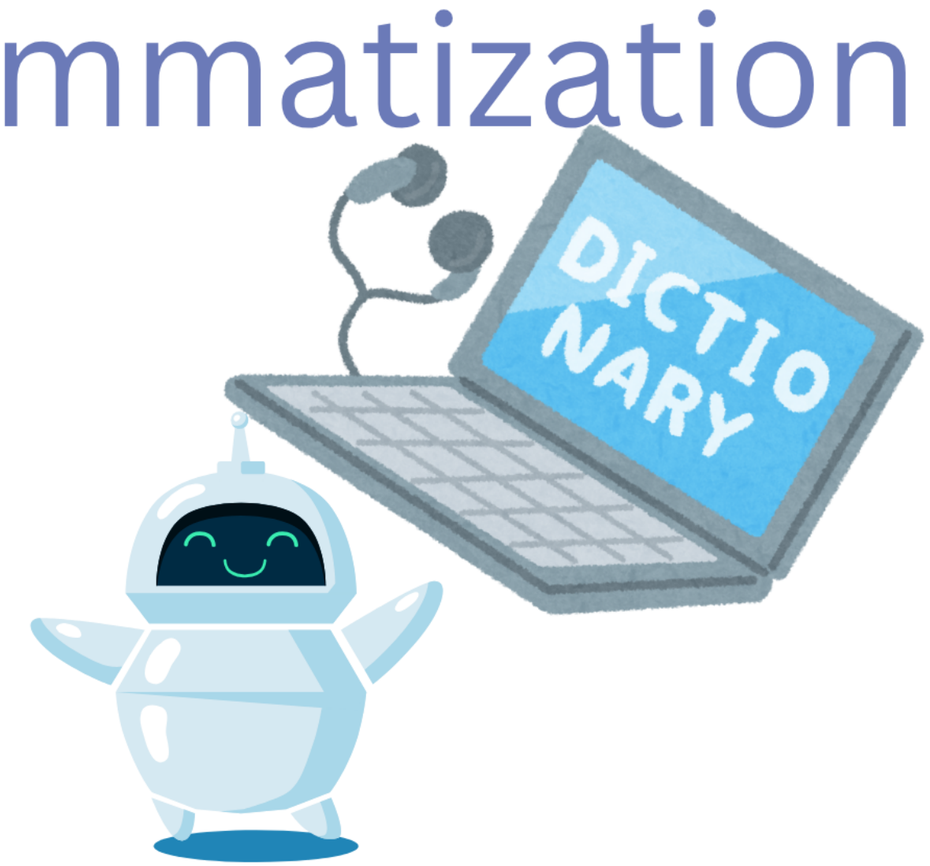
Stemming in code

```
from nltk.stem import PorterStemmer

stemmer = PorterStemmer()

tokens = ['running', 'bats', 'organizations', 'reading']

stemmed = [stemmer.stem(word) for word in tokens]

print(stemmed)
```

```
['run', 'bat', 'organizat', 'read']
```

Lemmatization

- Reduces words to their base form
- Ensures result is a valid word
  - organizations → organization
- Useful when grammatical accuracy is crucial
- Slower than stemming but more precise



Lemmatization in code

```
from nltk.stem import WordNetLemmatizer
nltk.download('wordnet')

lemmatizer = WordNetLemmatizer()

tokens = ['running', 'bats', 'organizations', 'reading']
lemmatized = [lemmatizer.lemmatize(word) for word in tokens]

print(lemmatized)
```

```
['running', 'bat', 'organization', 'reading']
```

Stemming

- Faster
- Results in non-dictionary words
- Use if needing speed

Lemmatization

- Slower
- More accurate
- Use if needing grammatical accuracy

Let's practice!

NATURAL LANGUAGE PROCESSING (NLP) IN PYTHON