Token classification

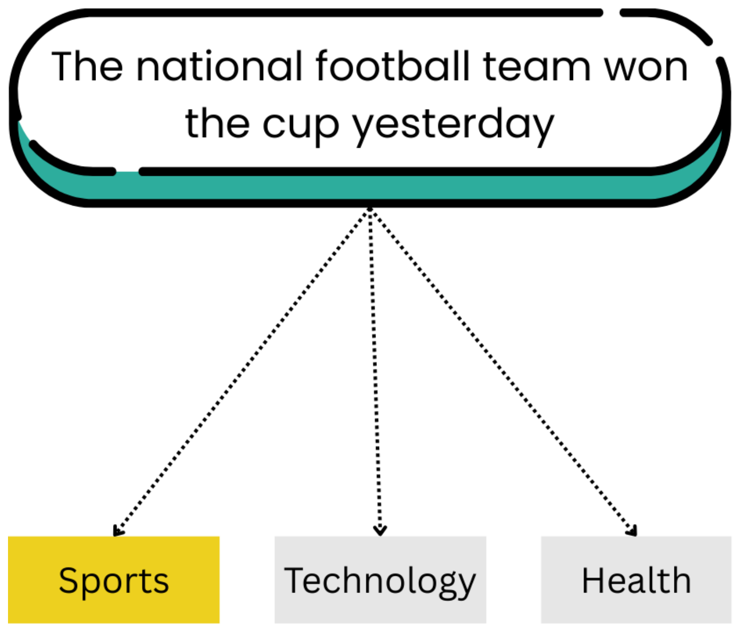
NATURAL LANGUAGE PROCESSING (NLP) IN PYTHON

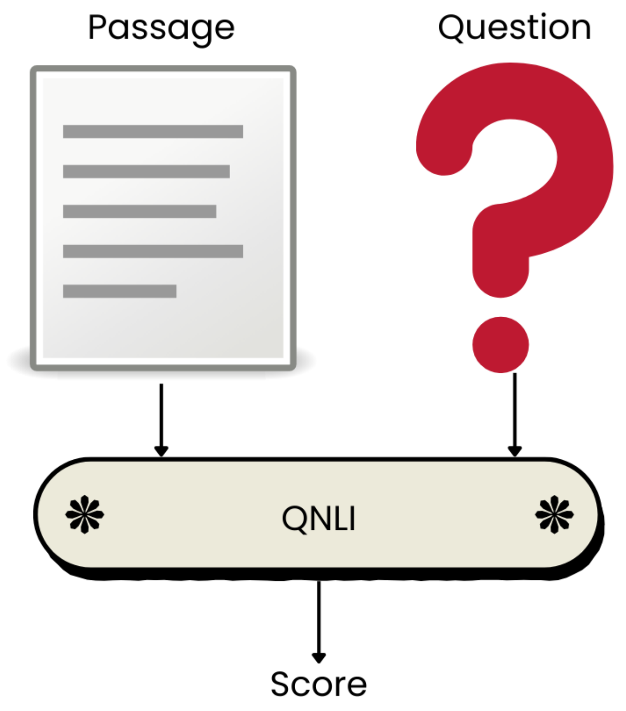
Fouad Trad
Machine Learning Engineer

Text versus token classification

Text classification

- Classifies entire sentences or pairs of texts

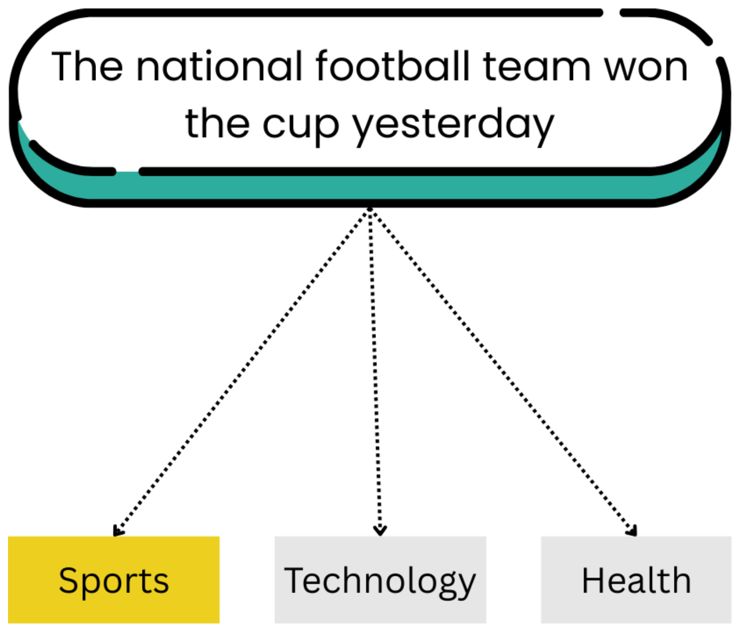


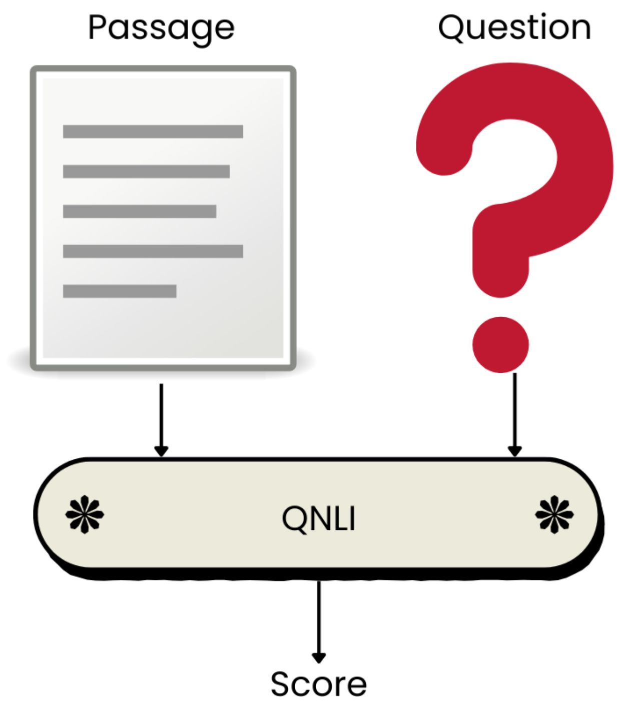


Text versus token classification

Text classification

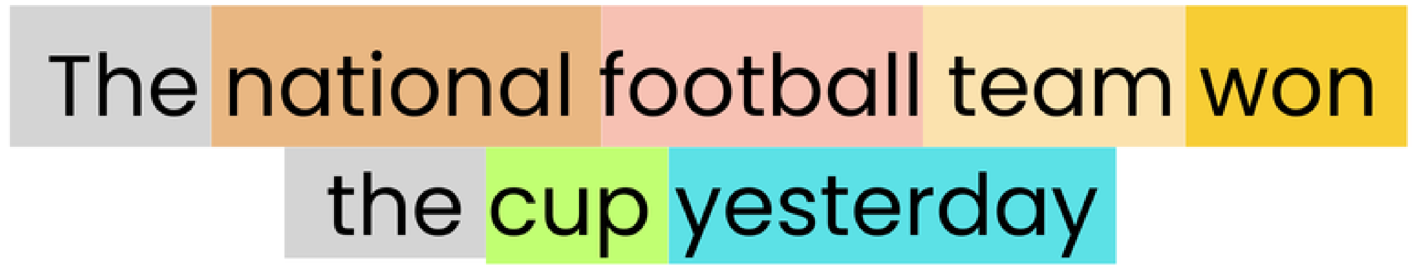
- Classifies entire sentences or pairs of texts





Token classification

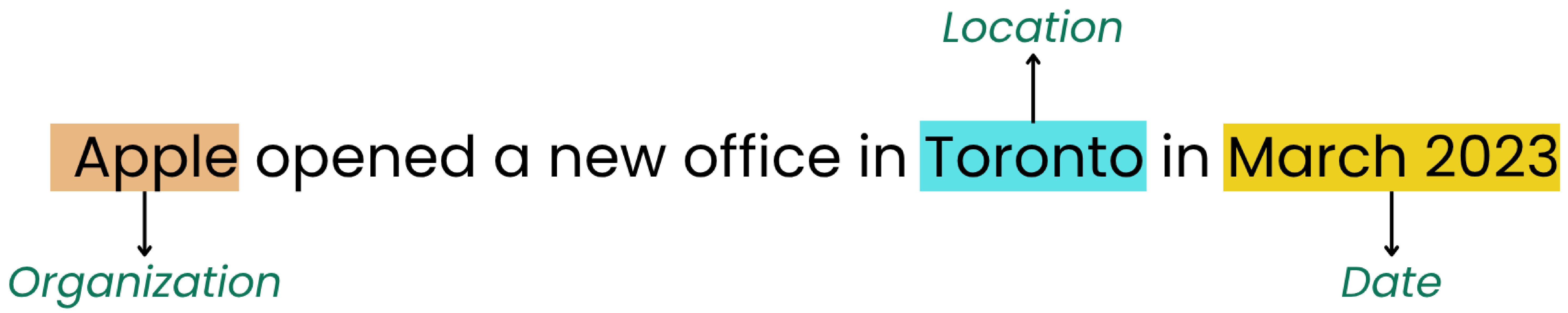
- Assigns labels to tokens within a sentence



- Named entity recognition (NER)
- Part of speech (PoS) tagging

Named entity recognition (NER)

- Identifies entities like names, locations, organizations, dates, and more



- Useful in:
  - Information retrieval
  - Question answering

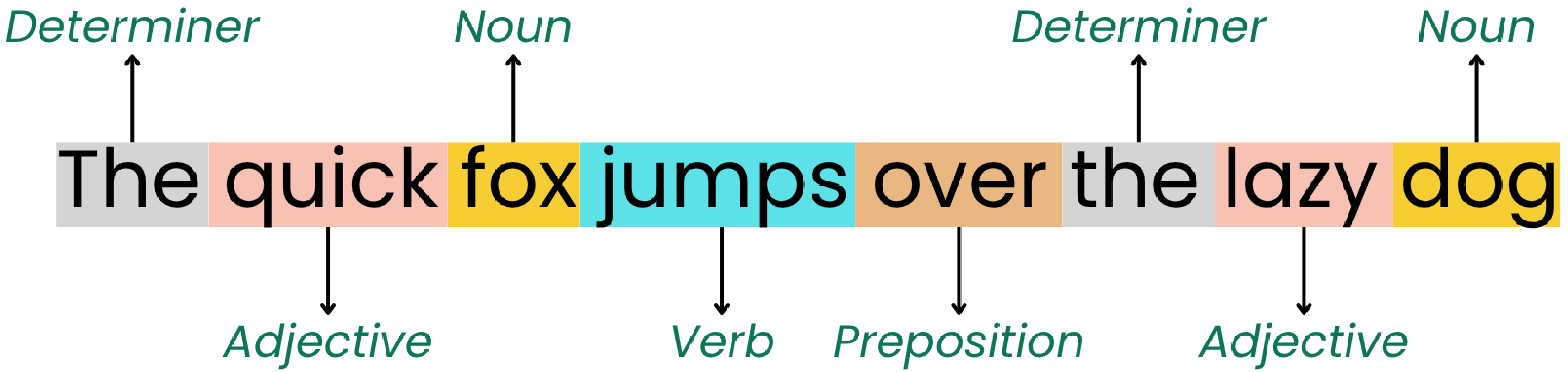
NER in code

```python
from transformers import pipeline
ner_pipeline = pipeline(task="ner",
                        model="dslim/bert-base-NER",
                        grouped_entities=True)

ner_results = ner_pipeline("Zara Venn established NovaCore Dynamics in London.")
print(ner_results)
```

```
[{'entity_group': 'PER', 'score': np.float32(0.99840075), 'word': 'Zara Venn', 'start': 0, 'end': 9},
 {'entity_group': 'ORG', 'score': np.float32(0.99875560), 'word': 'NovaCore Dynamics', 'start': 21, 'end': 38},
 {'entity_group': 'LOC', 'score': np.float32(0.99960726), 'word': 'London', 'start': 42, 'end': 48}]
```

Part of speech (PoS) tagging

- Assigns grammatical roles (noun, verb, adjective) to each word



- Useful in:
  - Syntactic parsing
  - Grammar correction
  - Text generation

PoS tagging in code

```
pos_pipeline = pipeline(task="token-classification",
                        model="vblagoje/bert-english-uncased-finetuned-pos",
                        grouped_entities=True)

pos_results = pos_pipeline("Zara Venn established NovaCore Dynamics in London.")
print(pos_results)
```

```
[{'entity_group': 'PROPN', 'score': np.float32(0.9982983), 'word': 'zara venn', 'start': 0, 'end': 9},
 {'entity_group': 'VERB', 'score': np.float32(0.99940944), 'word': 'established', 'start': 10, 'end': 21},
 {'entity_group': 'PROPN', 'score': np.float32(0.99455726), 'word': 'novacore dynamics', 'start': 22, 'end': 39},
 {'entity_group': 'ADP', 'score': np.float32(0.99935526), 'word': 'in', 'start': 40, 'end': 42},
 {'entity_group': 'PROPN', 'score': np.float32(0.99847955), 'word': 'london', 'start': 43, 'end': 49}]
```

Let's practice!

NATURAL LANGUAGE PROCESSING (NLP) IN PYTHON

Question answering

NATURAL LANGUAGE PROCESSING (NLP) IN PYTHON

Fouad Trad
Machine Learning Engineer

Extractive QA

- Answer directly copied from the passage

Context: The library closes at 6 PM on weekdays

Extractive Answer: 6 PM

Abstractive QA

- Model generates natural-sounding answer

Question: When does the library close on weekdays?

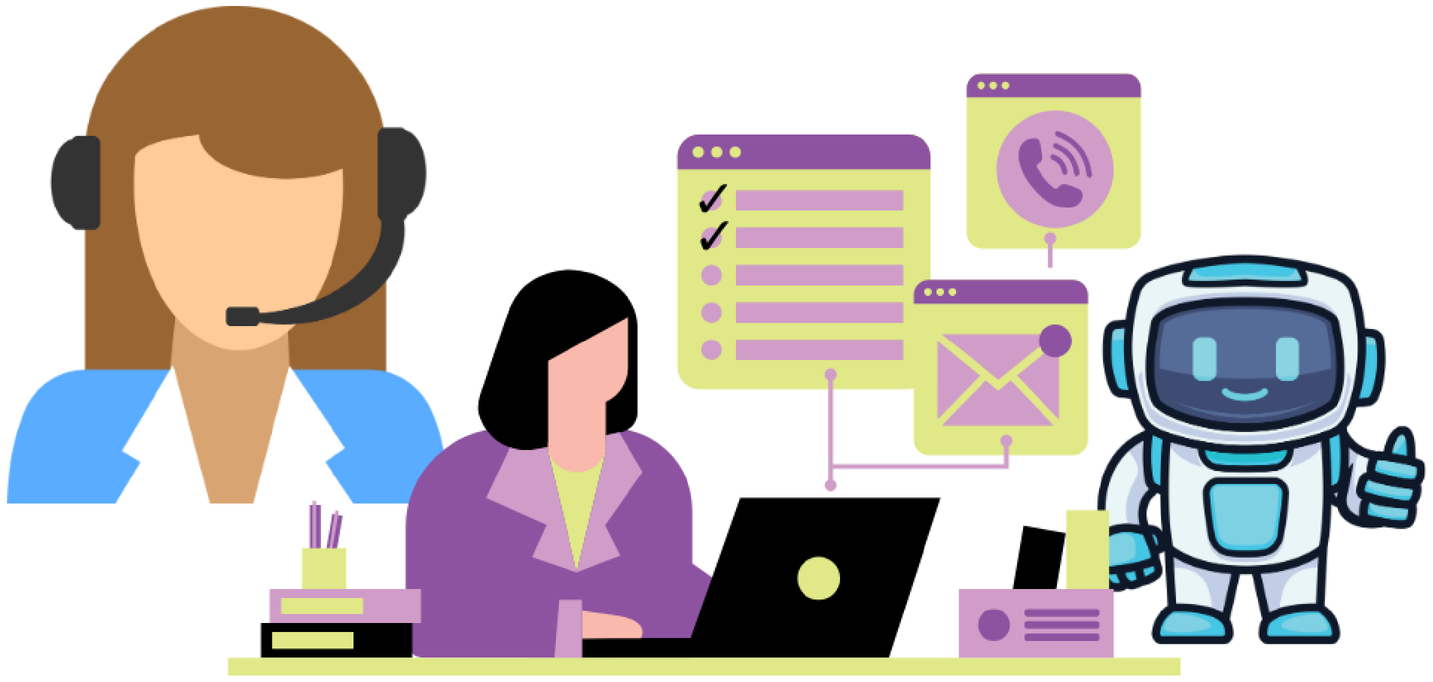
Abstractive Answer: Closure is at 6 PM.

Extractive QA

- Useful in:
  - Search engines
  - Document retrieval systems
  - Reading comprehension apps
- More accurate, less prone to errors

Abstractive QA

- Useful in:
  - Conversational agents
  - Virtual assistants
  - Customer support bots
- Can introduce errors



Extractive QA

```python
from transformers import pipeline
qa_pipeline = pipeline(task="question-answering",
                       model="distilbert/distilbert-base-cased-distilled-squad")
context = """The Amazon rainforest is the largest tropical rainforest in the world,
covering parts of Brazil, Peru, and Colombia."""
question = "Which countries does the Amazon rainforest cover?"
qa_answer = qa_pipeline(question=question, context=context)
print(qa_answer)
```

```
{'score': 0.9242347478866577,
 'start': 90, 'end': 116,
 'answer': 'Brazil, Peru, and Colombia'}
```

Abstractive QA

```
from transformers import pipeline
qa_pipeline = pipeline(task="text2text-generation",
                       model="fangyuan/hotpotqa_abstractive")
context = """The Amazon rainforest is the largest tropical rainforest in the world,
covering parts of Brazil, Peru, and Colombia."""
question = "Which countries does the Amazon rainforest cover?"
input_text = f"question: {question}  context: {context}"
result = qa_pipeline(input_text)
print(result)
```

```
[{'generated_text': 'The Amazon rainforest covers parts of Brazil, Peru, and Colombia.'}]
```

Let's practice!

NATURAL LANGUAGE PROCESSING (NLP) IN PYTHON

Sequence generation tasks

NATURAL LANGUAGE PROCESSING (NLP) IN PYTHON

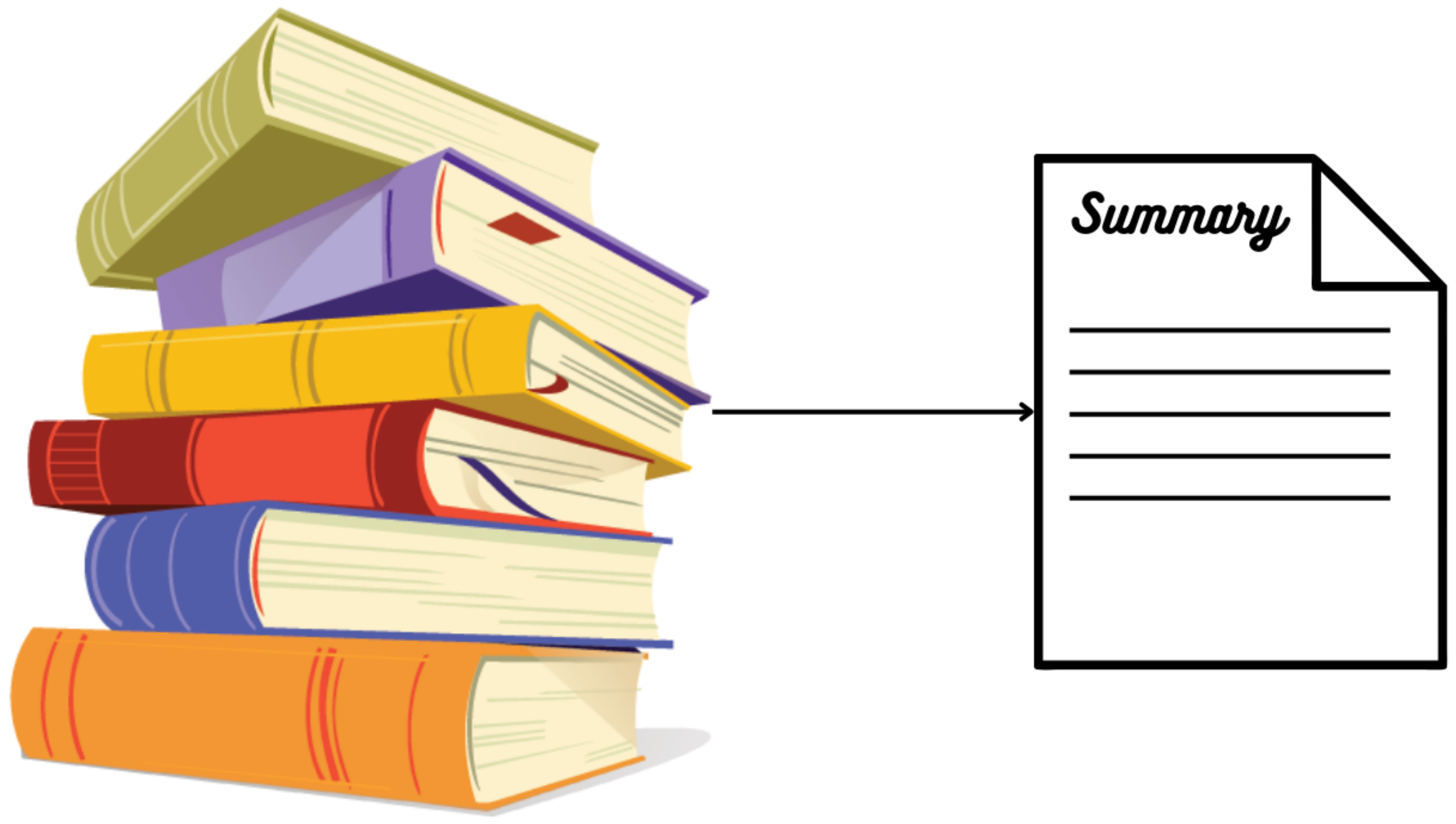
Fouad Trad
Machine Learning Engineer

Sequence generation

- Produces new text based on given input
- Includes tasks like:
  - Text summarization
  - Text translation
  - Language modeling

Text summarization

- Condenses long documents into shorter versions highlighting key points
- Useful when dealing with:
  - Lengthy news articles
  - Research papers
  - Reports
  - Emails



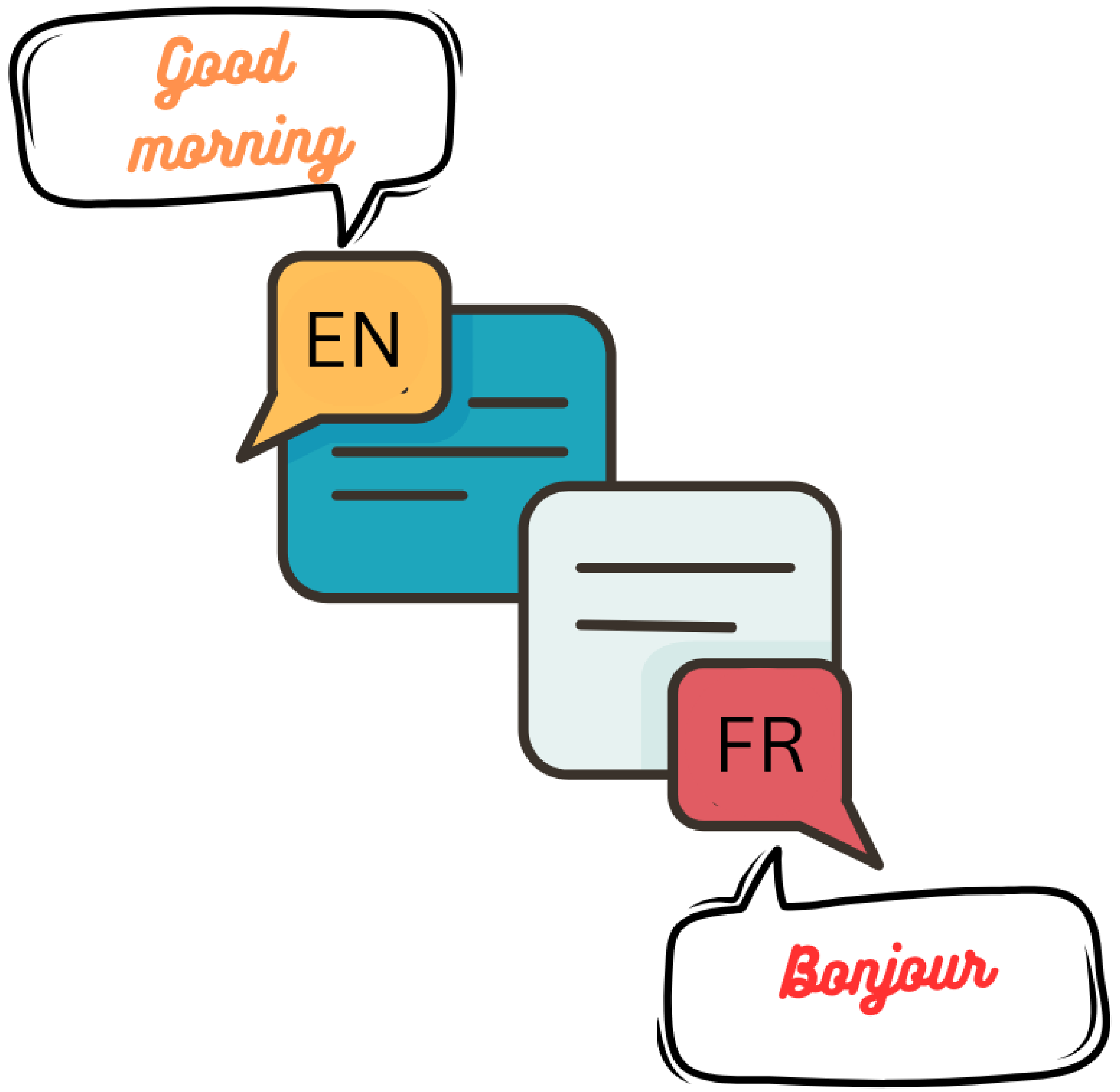
Text summarization pipeline

```python
from transformers import pipeline
summarizer = pipeline(task="summarization", model="cnicu/t5-small-booksum")

text = """The Amazon rainforest, often referred to as the "lungs of the Earth," is one of the most
biologically diverse regions in the world. Spanning over nine countries in South America, the majority
of the forest lies in Brazil. It is home to an estimated 390 billion individual trees, divided into
16,000 different species. The rainforest plays a critical role in regulating the global climate by
absorbing vast amounts of carbon dioxide and producing oxygen."""

result = summarizer(text)
print(result)
```

```
[{'summary_text': 'the Amazon rainforest is one of the most biologically diverse regions in the world.
The majority of the forest lies in Brazil. The rainforest plays a critical role in regulating the
global climate by absorbing vast amounts of carbon dioxide and producing oxygen.'}]
```

Text translation

- Converts text from one language to another
- Crucial in multilingual applications:
  - International websites
  - Customer support tools



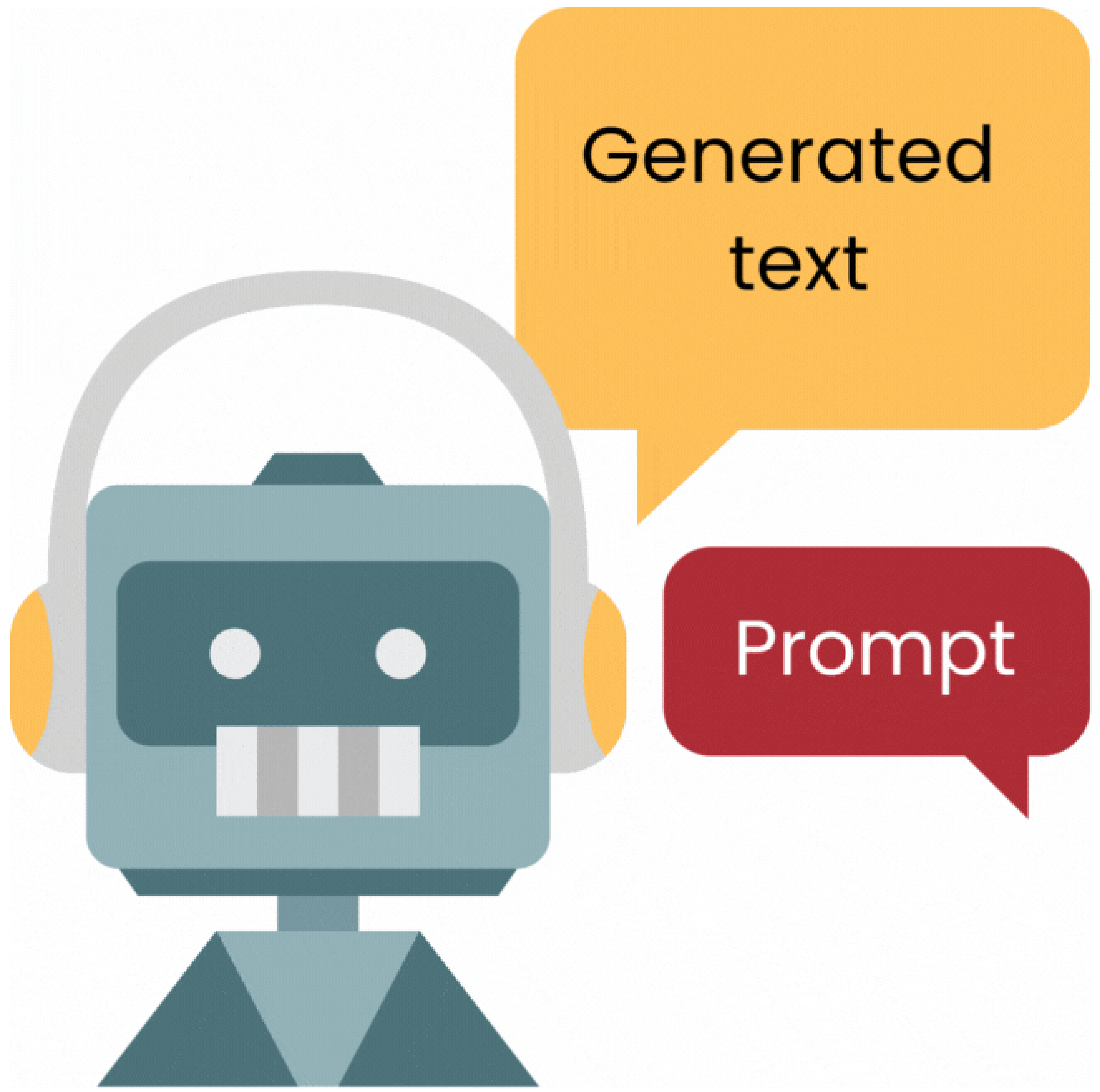
Text translation pipeline

```python
translator = pipeline(task="translation", model="Helsinki-NLP/opus-mt-en-fr")


sentence = "The rainforest helps regulate the Earth's climate."
result = translator(sentence)
print(result)
```

```
[{'translation_text': 'La forêt tropicale aide à réguler le climat de la Terre.'}]
```

Language modeling

- Predict the next words based on a given prompt
- Basis for many applications:
  - Autocompletion
  - Story generation
  - Chatbot replies



Language modeling pipeline

```
generator = pipeline(task="text-generation", model="distilgpt2")

prompt = "Once upon a time,"
result = generator(prompt, max_length=30, num_return_sequences=3)
print(result)
```

```
[{'generated_text': "Once upon a time, my life wasn't so good, I kept my things tidy. The
                     more time I spend with my children the more ..."},
 {'generated_text': 'Once upon a time, we began a process of finding the right answers to
                     some big questions," said Jim Pelterer, a lecturer at the
                     University...'},
 {'generated_text': 'Once upon a time, a man came along and took in the city, and found out
                     that a strange woman had just walked in and was dancing about...'}]
```

Let's practice!

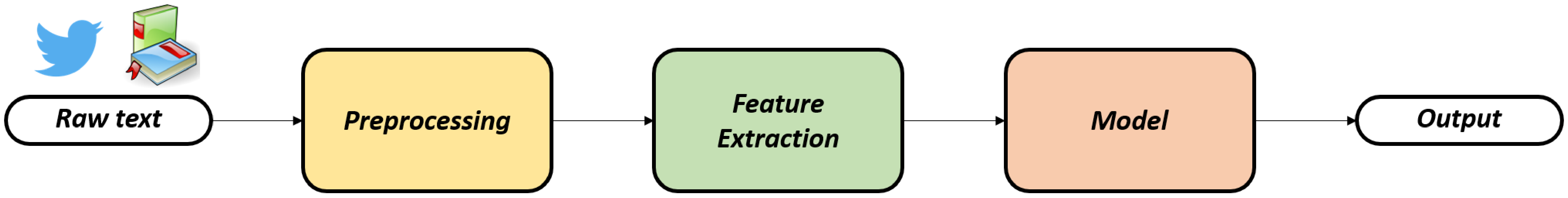
NATURAL LANGUAGE PROCESSING (NLP) IN PYTHON

Congratulations

NATURAL LANGUAGE PROCESSING (NLP) IN PYTHON

Fouad Trad
Machine Learning Engineer

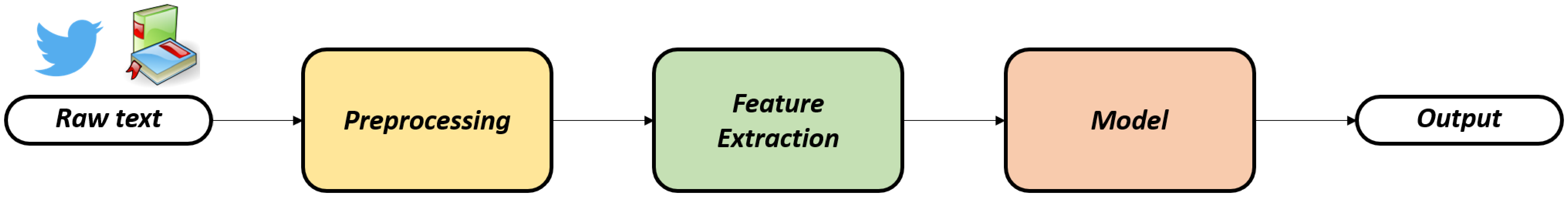
Chapter 1



Chapter 1

Tokenization
Cleaning
Normalization

Chapter 2



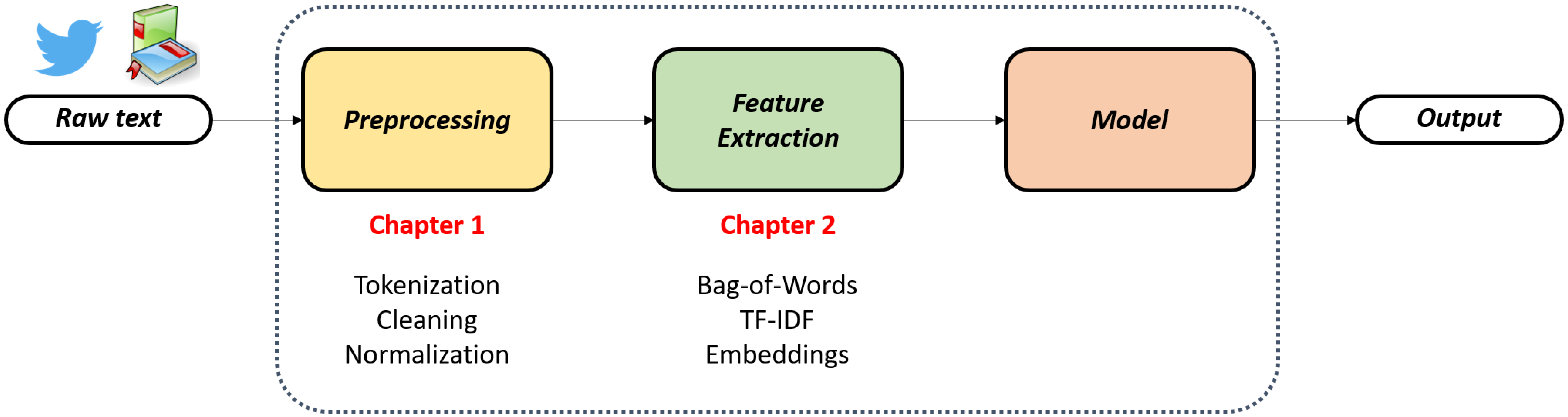
Chapter 1

Tokenization
Cleaning
Normalization

Chapter 2

Bag-of-Words
TF-IDF
Embeddings

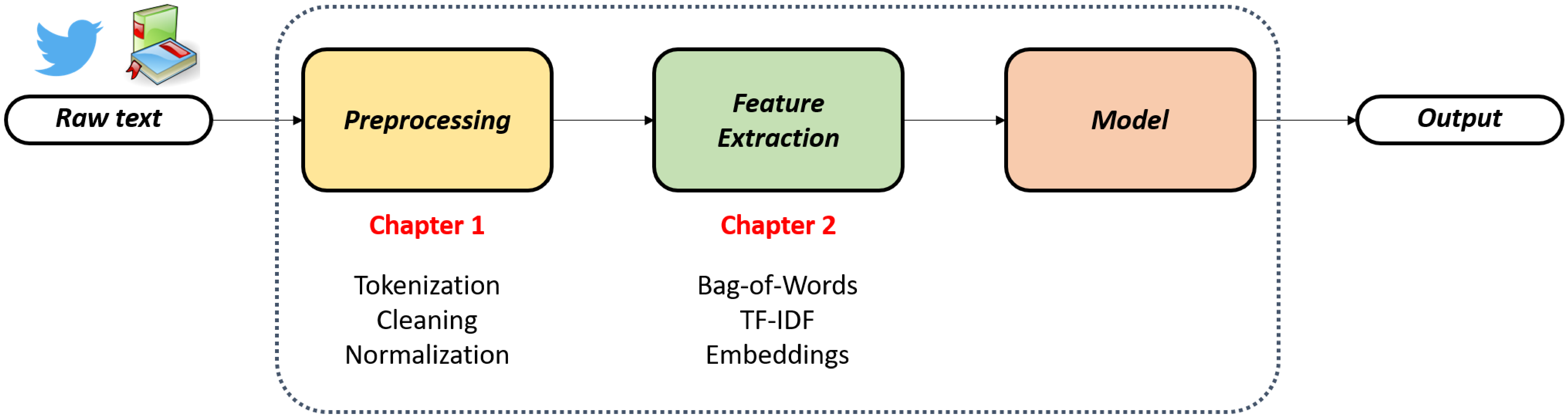
Chapter 3



Chapter 3

Sentiment analysis
Classifying content
Identifying paraphrases
Identifying grammar issues

Chapter 4



Chapter 3

Sentiment analysis
Classifying content
Identifying paraphrases
Identifying grammar issues

Chapter 4

Named Entity Recognition
Part of Speech Tagging
Question answering
Sequence generation

Congratulations!

NATURAL LANGUAGE PROCESSING (NLP) IN PYTHON