ing manual tokenization.

🎯 Goal: `pipeline` — Step by Step, All Tasks

> ✅ All examples are: **beginner-friendly**, **no manual tokenizer**, and **ready-to-run**

🧠 First: Install Transformers

pip install transformers

🛠️ General Syntax of pipeline:

from transformers import pipeline

pipe = pipeline(task_name, model="model_name")
pipe("Your input text here")

🔥 1. Sentiment Analysis

from transformers import pipeline

sentiment = pipeline("sentiment-analysis")
print(sentiment("I love this movie!"))

Custom Model:

sentiment = pipeline("sentiment-analysis",
model="distilbert-base-uncased-finetuned-sst-2-english")

🧠 2. Text Generation (GPT, GPT2)

generator = pipeline("text-generation", model="gpt2")

print(generator("Once upon a time", max_length=30, num_return_sequences=1))

More models:

- `"EleutherAI/gpt-neo-125M"`
- `"tiiuae/falcon-7b-instruct"` (requires GPU)

📄 3. Text Summarization (T5, BART)

summarizer = pipeline("summarization", model="facebook/bart-large-cnn")
print(summarizer("Long article or paragraph goes here...", max_length=50, min_length=20))

🌍 4. Translation (English → Other languages)

translator = pipeline("translation_en_to_fr", model="Helsinki-NLP/opus-mt-en-fr")
print(translator("How are you?"))

Other models:

- `"Helsinki-NLP/opus-mt-en-de"` (German)
- `"Helsinki-NLP/opus-mt-en-hi"` (Hindi)

❓ 5. Question Answering

qa = pipeline("question-answering")
print(qa({
  "question": "What is the capital of France?",

```
  "context": "France is a country in Europe. Its capital is Paris."
}))
```

🏷️ 6. Named Entity Recognition (NER)

```
ner = pipeline("ner", grouped_entities=True)
print(ner("My name is John and I live in New York City."))
```

🎙️ 7. Zero-Shot Classification

Classify text into labels **without training**:

```
classifier = pipeline("zero-shot-classification")
print(classifier(
    "This is a powerful laptop for gaming.",
    candidate_labels=["technology", "sports", "food"]
))
```

🔐 8. Fill in the Blanks (Masked Language Modeling)

```
fill_mask = pipeline("fill-mask", model="bert-base-uncased")
print(fill_mask("The capital of France is [MASK]."))
```

🖼️ 9. Image Classification (requires image)

```
from transformers import pipeline
from PIL import Image

image = Image.open("cat.jpg")  # Load any image
classifier = pipeline("image-classification", model="google/vit-base-patch16-224")
print(classifier(image))
```

🔤 10. Automatic Speech Recognition (Audio to Text)

from transformers import pipeline

asr = pipeline("automatic-speech-recognition", model="openai/whisper-base")
print(asr("path_to_audio_file.wav"))

📚 Bonus Models to Explore by Task

Task	Models
Text generation	`gpt2`, `gpt-neo`, `falcon`, `openai-community/gpt2-medium`
Summarization	`t5-small`, `facebook/bart-large-cnn`, `google/pegasus-xsum`
Translation	`Helsinki-NLP/opus-*`
QA	`distilbert-base-cased-distilled-squad`, `deepset/roberta-base-squad2`
NER	`dbmdz/bert-large-cased-finetuned-conll03-english`
Zero-shot	`facebook/bart-large-mnli`, `joeddav/xlm-roberta-large-xnli`

✅ Summary to Teach Students:

Task	Code
Sentiment	`pipeline("sentiment-analysis")`
Text Generation	`pipeline("text-generation")`
Summarization	`pipeline("summarization")`
Translation	`pipeline("translation_en_to_fr")`
QA	`pipeline("question-answering")`

NER	`pipeline("ner")`
Zero-shot	`pipeline("zero-shot-classification")`
Fill-mask	`pipeline("fill-mask")`
Image Classification	`pipeline("image-classification")`
ASR (speech)	`pipeline("automatic-speech-recognition")`

-