Slide 01

INTRODUCTION TO BERT

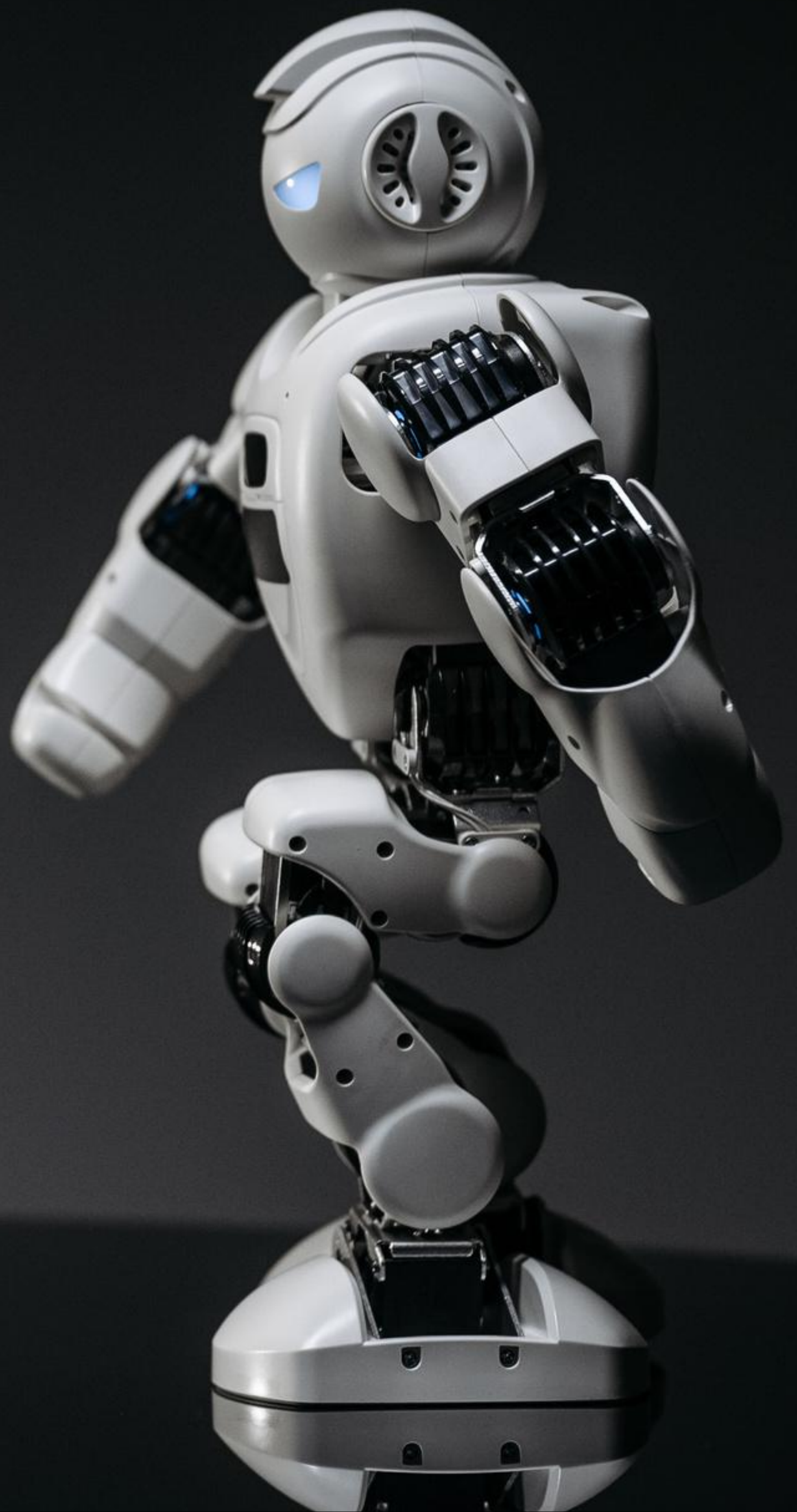
HASAN ABDELHADY

AIE - 400 - NLP

Slide 02

WHAT IS BERT?

BERT stands for **Bidirectional Encoder Representations from Transformers**. It’s a model created by Google AI in 2018 to help computers understand human language better. It was a big breakthrough in the field of Natural Language Processing (NLP), which is all about teaching computers to read, understand, and respond to human language.



What is BERT?

WHAT IS BERT?

BERT WAS TRAINED ON:

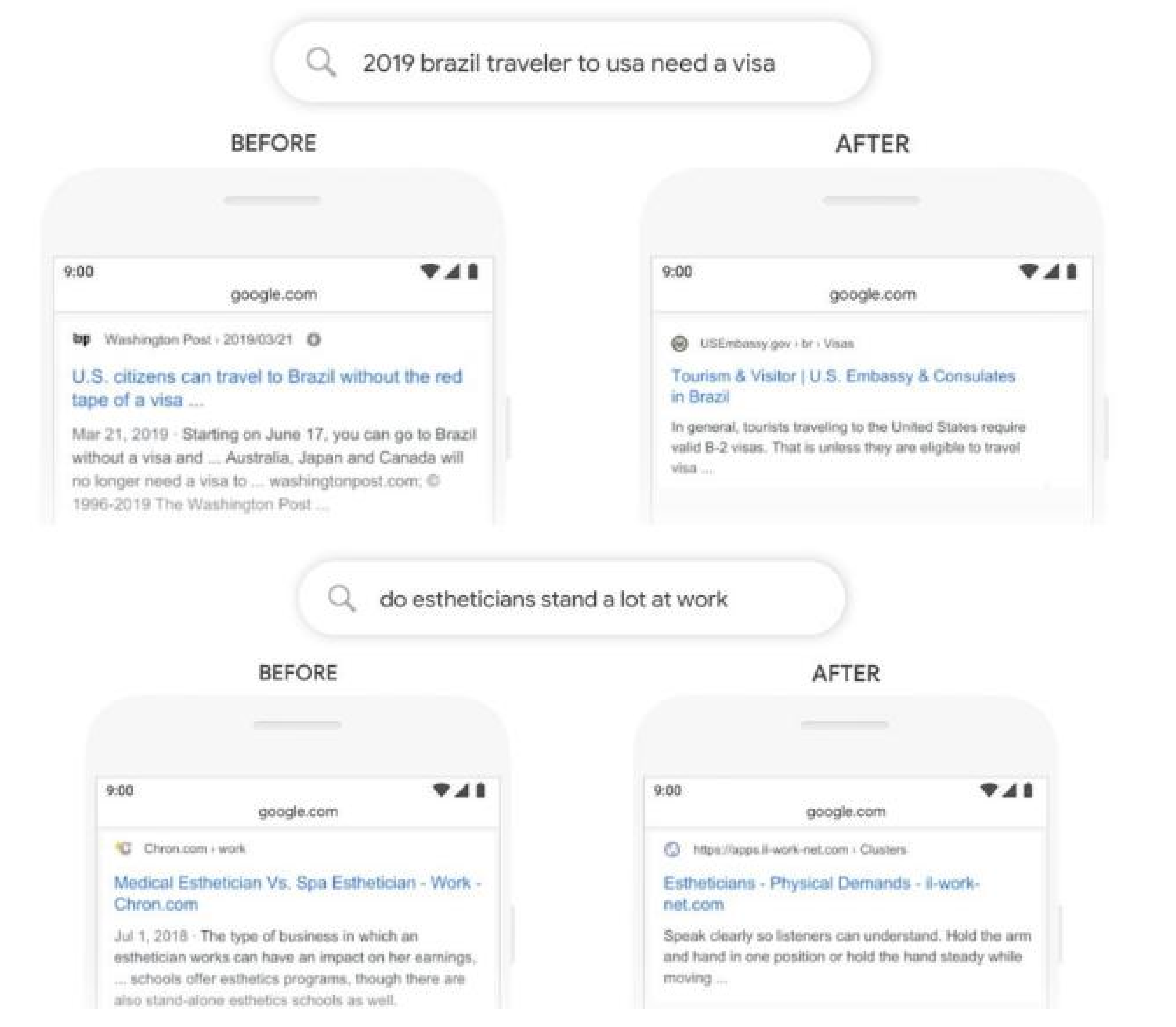
- THE ENTIRETY OF WIKIPEDIA
- 10s of thousands of books with a total of 3.3 Billion Words

Why BERT?

WHY BERT?

- Previous models read text one way (left-to-right or right-to-left)
- BERT reads text both ways for better context understanding
- Example: '**The bank is near the river**'
  - Left to right misinterpretation

How Does BERT Work?



2019 brazil traveler to usa need a visa

BEFORE

9:00

google.com

Washington Post › 2019/03/21

U.S. citizens can travel to Brazil without the red tape of a visa ...

Mar 21, 2019 · Starting on June 17, you can go to Brazil without a visa and ... Australia, Japan and Canada will no longer need a visa to ... washingtonpost.com; © 1996-2019 The Washington Post ...

AFTER

9:00

google.com

USEmbassy.gov › br › Visas

Tourism & Visitor | U.S. Embassy & Consulates in Brazil

In general, tourists traveling to the United States require valid B-2 visas. That is unless they are eligible to travel visa ...

do estheticians stand a lot at work

BEFORE

9:00

google.com

Chron.com › work

Medical Esthetician Vs. Spa Esthetician - Work - Chron.com

Jul 1, 2018 · The type of business in which an esthetician works can have an impact on her earnings. ... schools offer esthetics programs, though there are also stand-alone esthetics schools as well.

AFTER

9:00

google.com

https://apps.il-work-net.com › Clusters

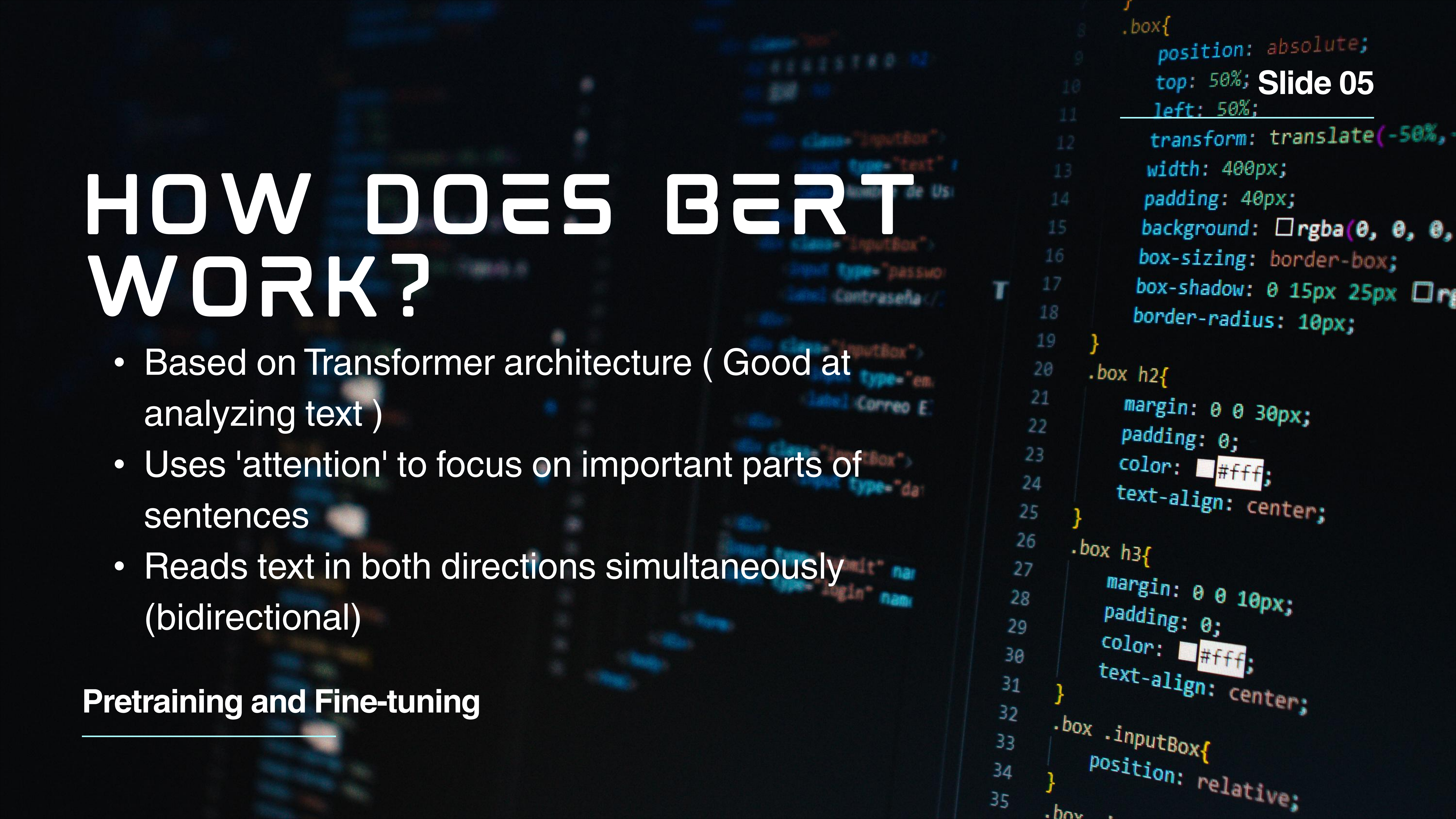
Estheticians - Physical Demands - il-work-net.com

Speak clearly so listeners can understand. Hold the arm and hand in one position or hold the hand steady while moving ...

HOW DOES BERT WORK?

- Based on Transformer architecture (Good at analyzing text)
- Uses 'attention' to focus on important parts of sentences
- Reads text in both directions simultaneously (bidirectional)

Pretraining and Fine-tuning



```
.box{
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%,
    width: 400px;
    padding: 40px;
    background: rgba(0, 0, 0,
    box-sizing: border-box;
    box-shadow: 0 15px 25px r
    border-radius: 10px;
}
.box h2{
    margin: 0 0 30px;
    padding: 0;
    color: #fff;
    text-align: center;
}
.box h3{
    margin: 0 0 10px;
    padding: 0;
    color: #fff;
    text-align: center;
}
.box .inputBox{
    position: relative;
}
.box
```

PRETRAINING AND FINE-TUNING

- Pretraining: Learns basic language patterns from large text data
- Fine-tuning: Adjusted for specific tasks (e.g., sentiment analysis)
- Adaptable to various NLP applications

Applications of BERT

APPLICATIONS OF BERT

- Question Answering
- Text Classification (e.g., spam detection)
- Named Entity Recognition (NER)
- Sentiment Analysis

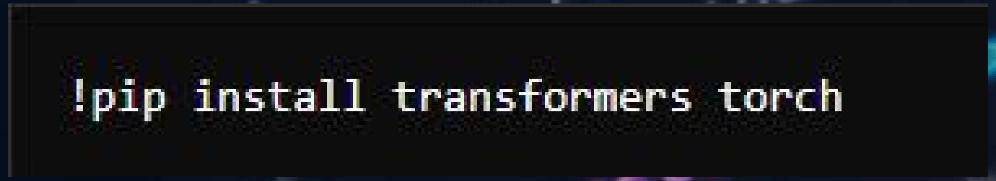
Why BERT is Special

WHY BERT IS SPECIAL

- Bidirectional reading enables deeper understanding
- Attention mechanism focuses on key words and relationships
- Pretrained on large datasets; adaptable to new tasks

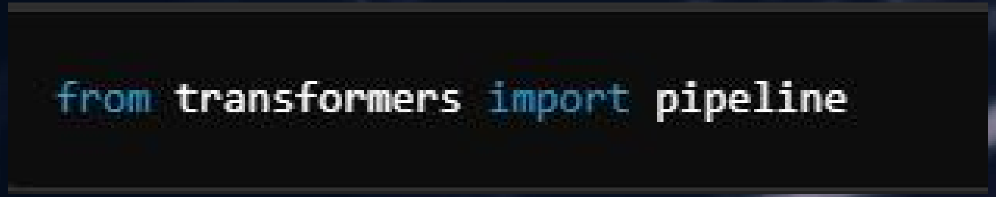
USING BERT IN YOUR CODEPACE

Installation



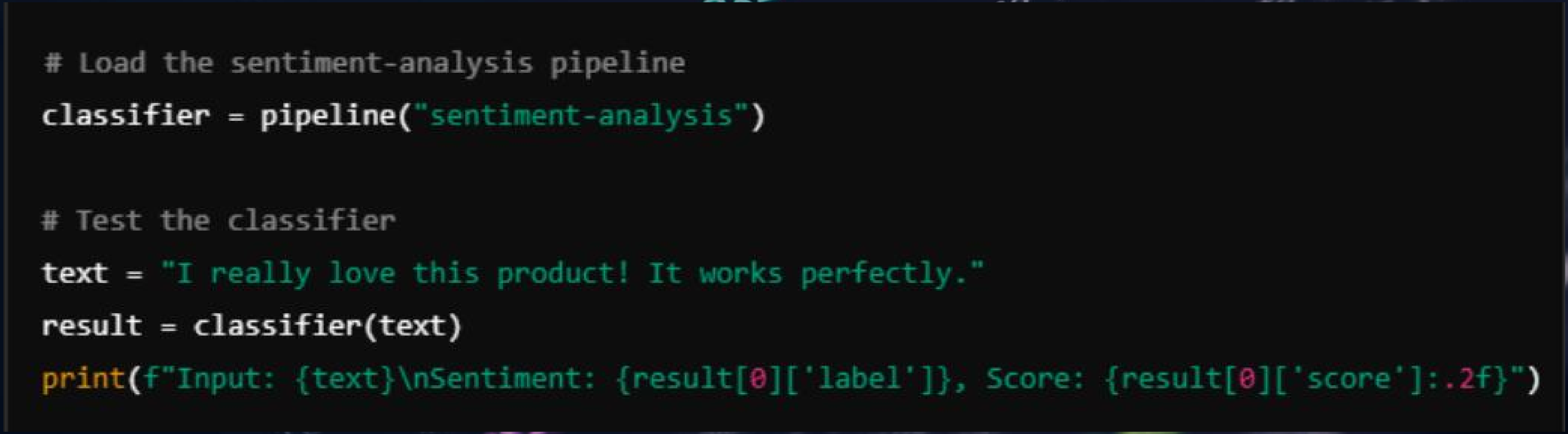
```
!pip install transformers torch
```

Import Libraries and Load BERT

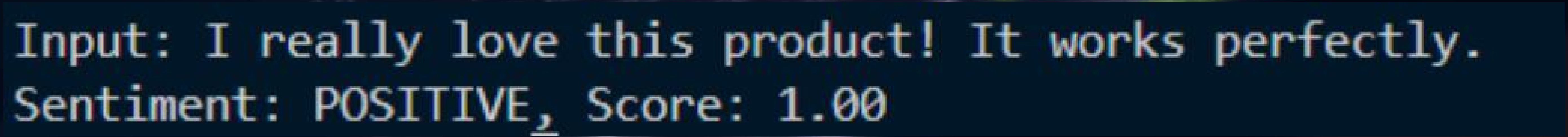


```
from transformers import pipeline
```

SENTIMENT ANALYSIS



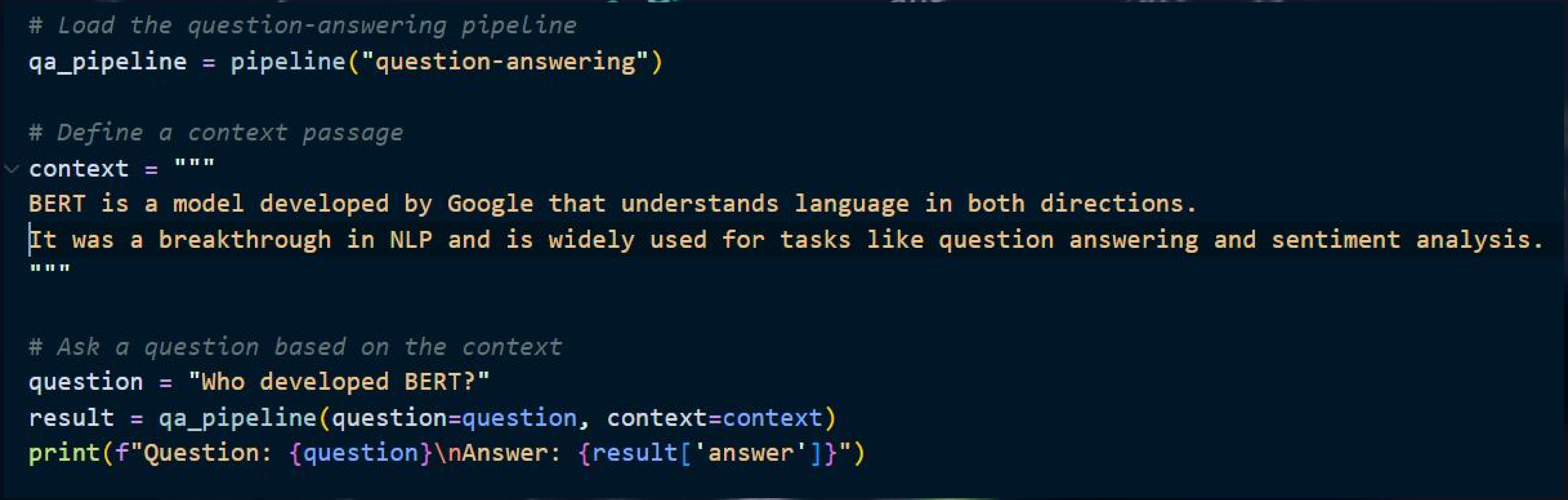
```python
# Load the sentiment-analysis pipeline
classifier = pipeline("sentiment-analysis")

# Test the classifier
text = "I really love this product! It works perfectly."
result = classifier(text)
print(f"Input: {text}\nSentiment: {result[0]['label']}, Score: {result[0]['score']:.2f}")
```



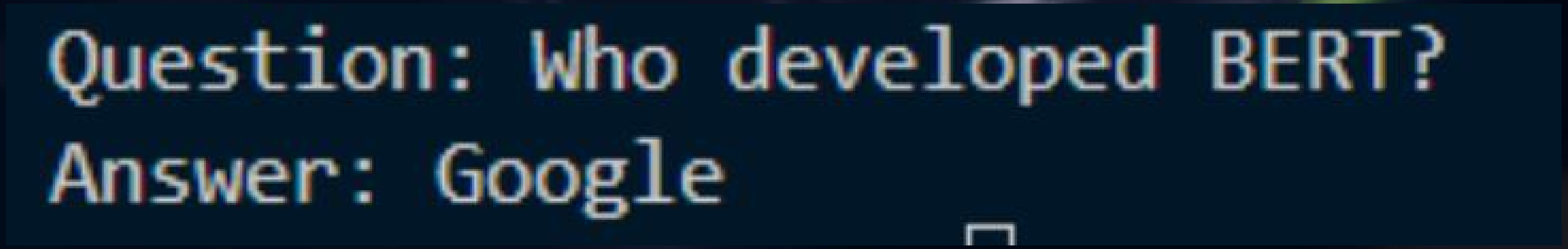
```
Input: I really love this product! It works perfectly.
Sentiment: POSITIVE, Score: 1.00
```

QUESTION ANSWRING

Slide 12

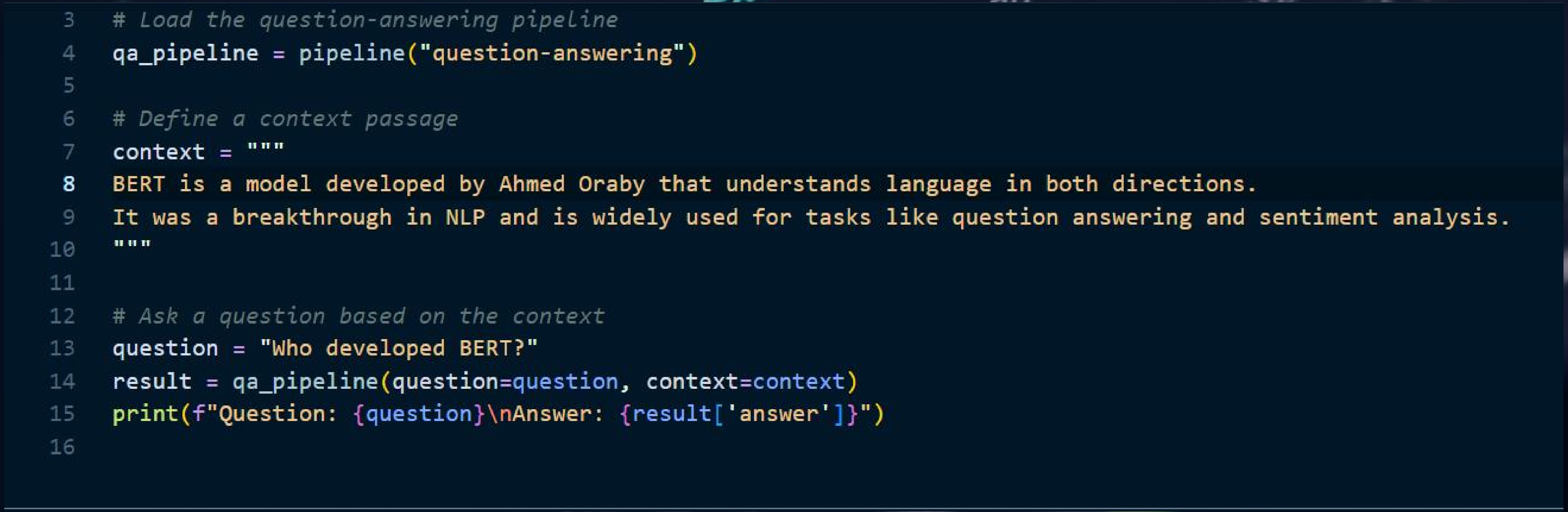


```python
# Load the question-answering pipeline
qa_pipeline = pipeline("question-answering")

# Define a context passage
context = """
BERT is a model developed by Google that understands language in both directions.
It was a breakthrough in NLP and is widely used for tasks like question answering and sentiment analysis.
"""


# Ask a question based on the context
question = "Who developed BERT?"
result = qa_pipeline(question=question, context=context)
print(f"Question: {question}\nAnswer: {result['answer']}")
```

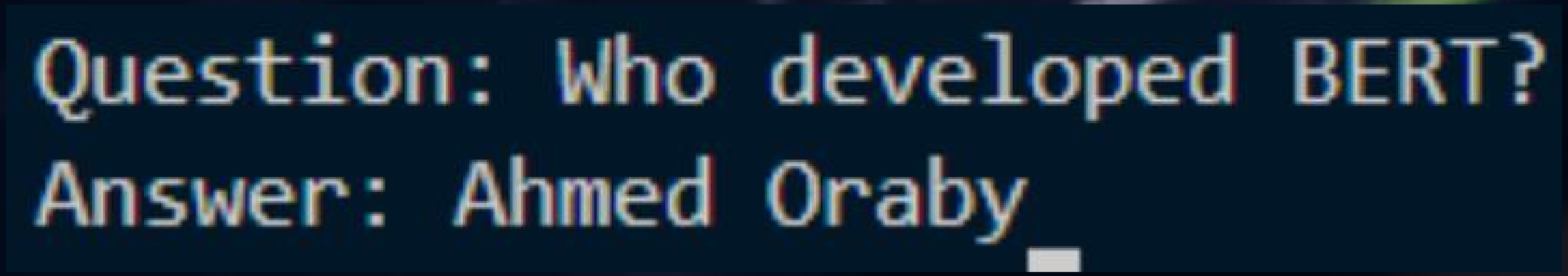


```
Question: Who developed BERT?
Answer: Google
```

QUESTION ANSWRING

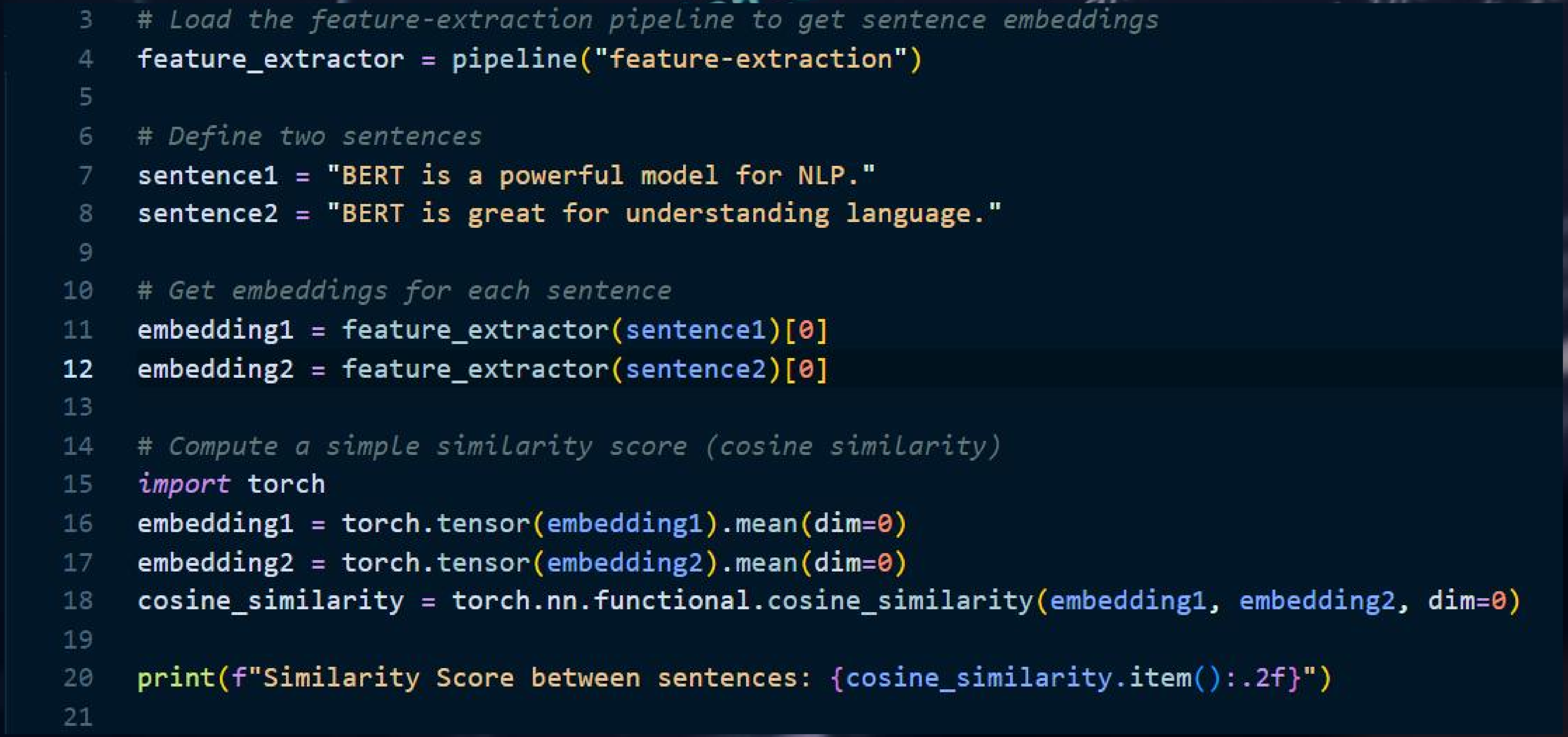


```python
# Load the question-answering pipeline
qa_pipeline = pipeline("question-answering")

# Define a context passage
context = """
BERT is a model developed by Ahmed Oraby that understands language in both directions.
It was a breakthrough in NLP and is widely used for tasks like question answering and sentiment analysis.
"""

# Ask a question based on the context
question = "Who developed BERT?"
result = qa_pipeline(question=question, context=context)
print(f"Question: {question}\nAnswer: {result['answer']}")

```

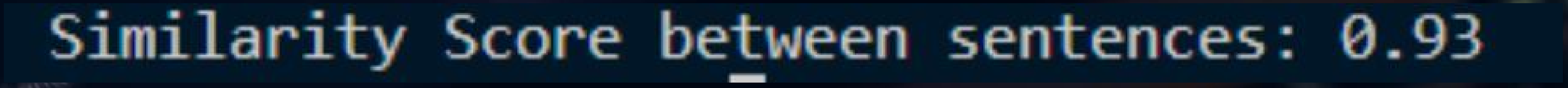


```
Question: Who developed BERT?
Answer: Ahmed Oraby
```

FEATURE SIMILARITY



```python
# Load the feature-extraction pipeline to get sentence embeddings
feature_extractor = pipeline("feature-extraction")

# Define two sentences
sentence1 = "BERT is a powerful model for NLP."
sentence2 = "BERT is great for understanding language."

# Get embeddings for each sentence
embedding1 = feature_extractor(sentence1)[0]
embedding2 = feature_extractor(sentence2)[0]

# Compute a simple similarity score (cosine similarity)
import torch
embedding1 = torch.tensor(embedding1).mean(dim=0)
embedding2 = torch.tensor(embedding2).mean(dim=0)
cosine_similarity = torch.nn.functional.cosine_similarity(embedding1, embedding2, dim=0)

print(f"Similarity Score between sentences: {cosine_similarity.item():.2f}")
```



```
Similarity Score between sentences: 0.93
```

THANK YOU

FOR YOUR *ATTENTION*