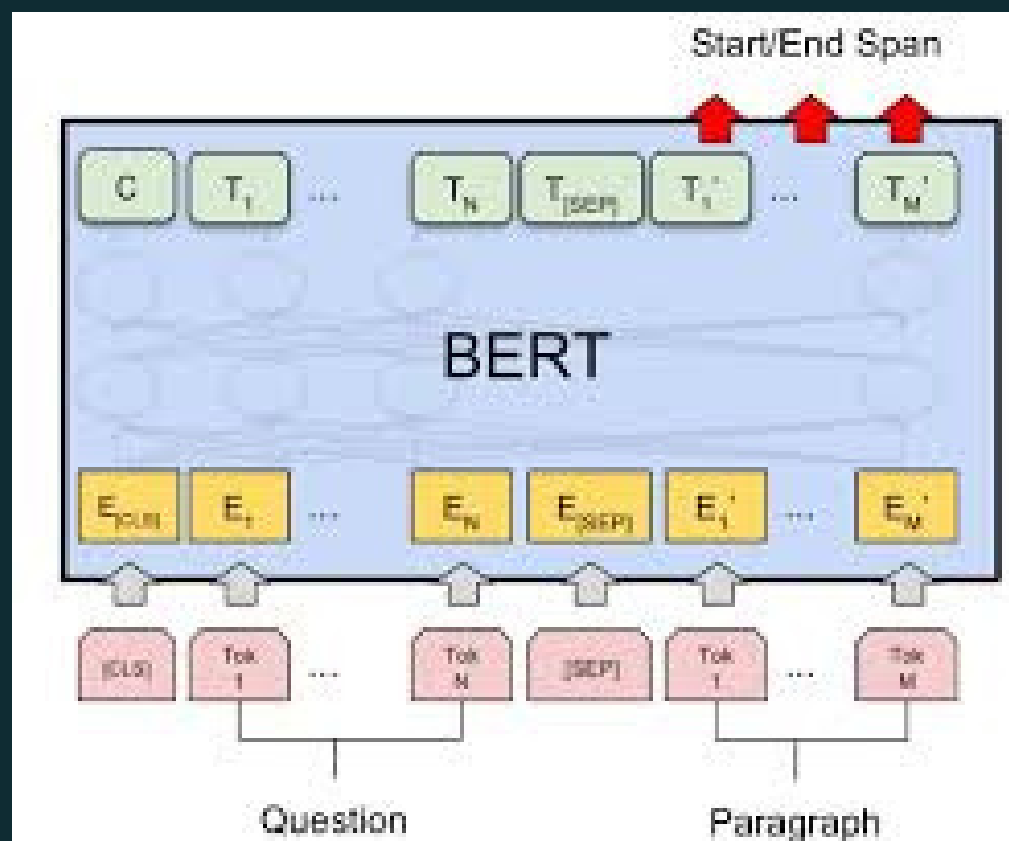


**What We
Need To
Know
About BERT!**

BERT

BERT stands for Bidirectional Encoder Representations from Transformers. It's a deep learning model developed by Google that has revolutionized the way machines understand and process human language.



Mr. Shivan Kumar

Data Scientist & Kaggle Master

Swipe >>>

How does BERT work?

BERT uses a technique called "transformer architecture" to analyze words in a sentence and understand their context by considering both the words before and after them. This bidirectional approach allows BERT to capture complex relationships and nuances within language, leading to remarkable accuracy in tasks like language translation, sentiment analysis, and question answering



Mr. Shivan Kumar

Data Scientist & Kaggle Master

Swipe >>>

Applications of BERT

Machine Translation: BERT's contextual understanding helps improve translation accuracy, making it a game-changer for multilingual communication.

Sentiment Analysis: BERT's ability to comprehend context enables more accurate sentiment analysis, aiding businesses in understanding customer feedback and trends.



Mr. Shivan Kumar

Data Scientist & Kaggle Master

Swipe >>>

Question Answering: BERT's contextual understanding allows it to answer questions based on given text, enhancing search engines and virtual assistants.

Fun Fact: BERT was trained on an astonishing amount of data, including a massive collection of books and a whole lot of web pages. Talk about a well-read AI!





```
from transformers import BertForQuestionAnswering,
BertTokenizer

# Load BERT pre-trained model and tokenizer
model_name = "bert-base-uncased"
model =
BertForQuestionAnswering.from_pretrained(model_name)
tokenizer = BertTokenizer.from_pretrained(model_name)

def answer_question(context, question):
    # Tokenize the context and question
    encoded_inputs = tokenizer.encode_plus(question,
context, add_special_tokens=True,
return_tensors="pt")

    # Retrieve the input IDs and attention mask
    input_ids = encoded_inputs["input_ids"]
    attention_mask = encoded_inputs["attention_mask"]
```



```
# Use the pre-trained model to predict the answer
outputs = model(input_ids,
attention_mask=attention_mask)
start_scores = outputs.start_logits
end_scores = outputs.end_logits

# Get the start and end indices of the answer
start_index = start_scores.argmax(dim=1).item()
end_index = end_scores.argmax(dim=1).item()

# Convert the token indices back to text
tokens =
tokenizer.convert_ids_to_tokens(input_ids.squeeze().t
olist())
answer =
tokenizer.convert_tokens_to_string(tokens[start_index
:end_index+1])
```



```
return answer
```

```
# Example usage
```

```
context = "BERT (Bidirectional Encoder  
Representations from Transformers) is a deep learning  
model developed by Google. It has revolutionized  
natural language processing (NLP) tasks by capturing  
complex relationships and nuances within language."
```

```
question = "What is BERT?"
```

```
answer = answer_question(context, question)
```

```
print("Question:", question)
```

```
print("Answer:", answer)
```

- Share this post with your fellow language enthusiasts and let's celebrate the power of **BERT** in shaping the future of NLP!

DID YOU FIND THIS POST HELPFUL ?

Follow us on



@engineer_bhaiya_yt



@engineeringwalabhaiya

topmate.io/shivan_kumar1

For Mentorship & Free Text
Queries,
WhatsApp at: +91 6280 963 446

Like, Save, and Share
with
your friends !!!



Shivan Kumar

Data Scientist & Kaggle Master

