

In [7]: `pip install transformers torch`

Requirement already satisfied: transformers in d:\ml\lib\site-packages (4.49.0)  
Requirement already satisfied: torch in d:\ml\lib\site-packages (2.3.1)  
Requirement already satisfied: filelock in d:\ml\lib\site-packages (from transformers) (3.13.1)  
Requirement already satisfied: huggingface-hub<1.0,>=0.26.0 in d:\ml\lib\site-packages (from transformers) (0.29.1)  
Requirement already satisfied: numpy>=1.17 in d:\ml\lib\site-packages (from transformers) (1.26.4)  
Requirement already satisfied: packaging>=20.0 in d:\ml\lib\site-packages (from transformers) (23.1)  
Requirement already satisfied: pyyaml>=5.1 in d:\ml\lib\site-packages (from transformers) (6.0.1)  
Requirement already satisfied: regex!=2019.12.17 in d:\ml\lib\site-packages (from transformers) (2023.10.3)  
Requirement already satisfied: requests in d:\ml\lib\site-packages (from transformers) (2.32.3)  
Requirement already satisfied: tokenizers<0.22,>=0.21 in d:\ml\lib\site-packages (from transformers) (0.21.0)  
Requirement already satisfied: safetensors>=0.4.1 in d:\ml\lib\site-packages (from transformers) (0.5.2)  
Requirement already satisfied: tqdm>=4.27 in d:\ml\lib\site-packages (from transformers) (4.67.1)  
Requirement already satisfied: typing-extensions>=4.8.0 in d:\ml\lib\site-packages (from torch) (4.12.2)  
Requirement already satisfied: sympy in d:\ml\lib\site-packages (from torch) (1.12)  
Requirement already satisfied: networkx in d:\ml\lib\site-packages (from torch) (3.1)  
Requirement already satisfied: jinja2 in d:\ml\lib\site-packages (from torch) (3.1.3)  
Requirement already satisfied: fsspec in d:\ml\lib\site-packages (from torch) (2023.10.0)  
Requirement already satisfied: mkl<=2021.4.0,>=2021.1.1 in d:\ml\lib\site-packages (from torch) (2021.4.0)  
Requirement already satisfied: intel-openmp==2021.\* in d:\ml\lib\site-packages (from mkl<=2021.4.0,>=2021.1.1->torch) (2021.4.0)  
Requirement already satisfied: tbb==2021.\* in d:\ml\lib\site-packages (from mkl<=2021.4.0,>=2021.1.1->torch) (2021.13.0)  
Requirement already satisfied: colorama in d:\ml\lib\site-packages (from tqdm>=4.27->transformers) (0.4.6)  
Requirement already satisfied: MarkupSafe>=2.0 in d:\ml\lib\site-packages (from jinja2->torch) (2.1.3)  
Requirement already satisfied: charset-normalizer<4,>=2 in d:\ml\lib\site-packages (from requests->transformers) (2.0.4)  
Requirement already satisfied: idna<4,>=2.5 in d:\ml\lib\site-packages (from requests->transformers) (3.4)  
Requirement already satisfied: urllib3<3,>=1.21.1 in d:\ml\lib\site-packages (from requests->transformers) (2.0.7)  
Requirement already satisfied: certifi>=2017.4.17 in d:\ml\lib\site-packages (from requests->transformers) (2024.2.2)  
Requirement already satisfied: mpmath>=0.19 in d:\ml\lib\site-packages (from sympy->torch) (1.3.0)  
Note: you may need to restart the kernel to use updated packages.

In [8]: `import torch  
from transformers import BertTokenizer, BertForSequenceClassification, BertForQuestionAnswering  
from transformers import pipeline  
  
# Load pre-trained BERT tokenizer and model  
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')  
model_sentiment = BertForSequenceClassification.from_pretrained('nlptown/bert-base-multilingual-uncased-sentiment') # Multilingual Sentiment Analysis  
model_qa = BertForQuestionAnswering.from_pretrained('bert-large-uncased-whole-word-masking-finetuned-squad')  
  
# Sentiment Analysis using BERT  
def analyze_sentiment(text):  
 inputs = tokenizer(text, return_tensors='pt', truncation=True, padding=True)  
 outputs = model_sentiment(**inputs)  
 logits = outputs.logits  
 sentiment = torch.argmax(logits).item()  
 sentiment_map = {  
 0: 'Very Negative',  
 1: 'Negative',  
 2: 'Neutral',  
 3: 'Positive',  
 4: 'Very Positive'  
 }  
 return sentiment_map[sentiment]  
  
# Text Classification (for example: Sentiment Analysis)  
def classify_text(text):  
 sentiment = analyze_sentiment(text)  
 return sentiment  
  
# Question Answering with BERT  
def answer_question(question, context):  
 inputs = tokenizer(question, context, return_tensors='pt')  
 outputs = model_qa(**inputs)  
 answer_start = torch.argmax(outputs.start_logits)  
 answer_end = torch.argmax(outputs.end_logits) + 1  
 answer = tokenizer.convert_tokens_to_string(tokenizer.convert_ids_to_tokens(inputs['input_ids'][0][answer_start:answer_end]))  
 return answer  
  
# Task 1: Sentiment Analysis  
def sentiment_analysis(text):  
 sentiment = analyze_sentiment(text)  
 return sentiment  
  
# Task 2: Question Answering  
def question_answering(question, context):  
 answer = answer_question(question, context)  
 return answer  
  
# --- Example of usage ---  
if __name__ == "__main__":  
 # Task 1: Sentiment Analysis  
 text = "I love programming in Python! It is very fun and powerful."  
 sentiment = sentiment_analysis(text)  
 print("Sentiment Analysis:", sentiment)  
  
 # Task 2: Question Answering  
 context = ""BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based model designed for various natural language processing tasks.  
 It is pre-trained on a large corpus of text and fine-tuned for specific tasks like question answering and sentiment analysis.""  
 question = "What is BERT?"  
 answer = question_answering(question, context)  
 print("\nQuestion Answering:", answer)`

config.json: 0%| | 0.00/953 [00:00<?, ?B/s]

D:\ml\Lib\site-packages\huggingface\_hub\file\_download.py:142: UserWarning: `huggingface\_hub` cache-system uses symlinks by default to efficiently store duplicated files but your machine does not support them in C:\Users\amith\cache\huggingface\hub\models--nlptown--bert-base-multilingual-uncased-sentiment. Caching files will still work but in a degraded version that might require more space on your disk. This warning can be disabled by setting the `HF\_HUB\_DISABLE\_SYMLINKS\_WARNING` environment variable. For more details, see [https://huggingface.co/docs/huggingface\\_hub/how-to-cache#limitations](https://huggingface.co/docs/huggingface_hub/how-to-cache#limitations).  
To support symlinks on Windows, you either need to activate Developer Mode or to run Python as an administrator. In order to activate developer mode, see this article: <https://docs.microsoft.com/en-us/windows/apps/get-started/enable-your-device-for-development>  
warnings.warn(message)

model.safetensors: 0%| | 0.00/669M [00:00<?, ?B/s]  
config.json: 0%| | 0.00/443 [00:00<?, ?B/s]

D:\ml\Lib\site-packages\huggingface\_hub\file\_download.py:142: UserWarning: `huggingface\_hub` cache-system uses symlinks by default to efficiently store duplicated files but your machine does not support them in C:\Users\amith\cache\huggingface\hub\models--bert-large-uncased-whole-word-masking-finetuned-squad. Caching files will still work but in a degraded version that might require more space on your disk. This warning can be disabled by setting the `HF\_HUB\_DISABLE\_SYMLINKS\_WARNING` environment variable. For more details, see [https://huggingface.co/docs/huggingface\\_hub/how-to-cache#limitations](https://huggingface.co/docs/huggingface_hub/how-to-cache#limitations).  
To support symlinks on Windows, you either need to activate Developer Mode or to run Python as an administrator. In order to activate developer mode, see this article: <https://docs.microsoft.com/en-us/windows/apps/get-started/enable-your-device-for-development>  
warnings.warn(message)

model.safetensors: 0%| | 0.00/1.34G [00:00<?, ?B/s]

Some weights of the model checkpoint at bert-large-uncased-whole-word-masking-finetuned-squad were not used when initializing BertForQuestionAnswering: ['bert.pooler.dense.bias', 'bert.pooler.dense.weight']  
- This IS expected if you are initializing BertForQuestionAnswering from the checkpoint of a model trained on another task or with another architecture (e.g. initializing a BertForSequenceClassification model from a BertForPreTraining model).  
- This IS NOT expected if you are initializing BertForQuestionAnswering from the checkpoint of a model that you expect to be exactly identical (initializing a BertForSequenceClassification model from a BertForSequenceClassification model).  
Sentiment Analysis: Neutral

Question Answering: bidirectional encoder representations from transformers

