GA-NLP Final Project: Customer Review Classification & Summarization

Image generated with Playground-v2.5 on Poe

Business Context

In today's digital era, **customer reviews are pivotal for e-commerce platforms and online service providers**, serving as a barometer for customer satisfaction, highlighting areas for improvement, and guiding business decisions. The intricate process of Aspect-based Sentiment Analysis is instrumental in dissecting these reviews, offering a granular understanding of customer sentiments towards various facets of a product or service. This nuanced approach enables businesses to pinpoint specific elements—be it the screen, keyboard, or customer service of a laptop—that may need enhancement.

Moreover, the advent of Large Language Models (LLMs) has revolutionized the way businesses approach the summarization of customer reviews and case briefs. By leveraging LLMs, companies can swiftly and accurately distill the essence of customer feedback, streamlining the process of understanding overall sentiments and facilitating the refinement of product offerings. This swift synthesis not only conserves time and resources but also plays a critical role in enhancing customer satisfaction, driving sales, and boosting revenue.

Additionally, this project will delve into the **fine-tuning of models to elevate the quality of summaries provided by LLMs**. This involves training models on domain-specific data to enhance their ability to generate more precise and relevant summaries, thus improving the utility of these summaries in real-world applications.

Embarking on this final project on Customer Review Aspect-based Classification & Summarization, coupled with an emphasis on model fine-tuning for enhanced summarization, equips you with invaluable skills applicable to real-world business contexts. Through hands-on experience with code and implementation specifics, you'll gain the proficiency to construct such solutions using open-source LLMs. This experience will not

only serve as a compelling Proof-of-Concept but also pave the way for the deployment and productionization of these cutting-edge solutions in business environments.

Project Objective

We will use a Large Language Model to **automate the Aspect-based Classification and Summarization of Customer Reviews** received by a business. The project will aim to predict review sentiment, aspect-based sentiment and generate summaries of these customer reviews, and will evaluate the performance of the LLM at these tasks through various evaluation schemes.

Question 1: Install the Necessary Libraries (4 Marks)

Important Note

This project is organized into three distinct sections:

1. Aspect-Based Sentiment Classification (24 Marks)
2. Parameter-Efficient Fine-Tuning and Evaluation (26 Marks)
3. Observations and Business Perspective (10 Marks)

Each section can be completed independently, so you DO NOT need to run the entire notebook every time.

Important Instructions

You have a total of 8 hours of lab usage time to complete the project. This time cannot be extended.

If you need a break, you can pause the labs and resume later.

The labs will automatically pause after 20 minutes of inactivity, as shown in figure 5.

To use your lab time effectively, download the notebook first and review the code before using your lab hours.

Part 1: Aspect Based Sentiment Classification

1.1 Installation and Import (4 Marks)

(A) Install necessary libraries (2 Marks)

(B) Import necessary libraries (2 Marks)

1.1 (A) Install necessary libraries (2 Marks)

In [ ]:
```
import torch
```

In [ ]:
```
# This part of code will skip all the un-necessary warnings which can occur during the execution of this project.
import warnings
warnings.filterwarnings("ignore", category=Warning)
```

In [ ]:
```
# Installation for GPU llama-cpp-python==0.2.28
!CMAKE_ARGS="-DLLAMA_CUDA=on" pip install llama-cpp-python==0.2.69
!pip install -q --no-deps "xformers<0.0.27" "trl<0.9.0" peft==0.12.0 accelerate==0.32.1 bitsandbytes==0.43.2
!pip install rich
!pip install tqdm
# Installation for the huggingface-hub
!pip install huggingface-hub
# installation for the datasets
!pip install datasets
```

```
Requirement already satisfied: llama-cpp-python==0.2.69 in /usr/local/lib/python3.10/dist-packages (0.2.69)
Requirement already satisfied: jinja2>=2.11.3 in /usr/local/lib/python3.10/dist-packages (from llama-cpp-python==0.2.69) (3.1.4)
Requirement already satisfied: diskcache>=5.6.1 in /usr/local/lib/python3.10/dist-packages (from llama-cpp-python==0.2.69) (5.6.
3)
Requirement already satisfied: typing-extensions>=4.5.0 in /usr/local/lib/python3.10/dist-packages (from llama-cpp-python==0.2.6
9) (4.12.2)
Requirement already satisfied: numpy>=1.20.0 in /usr/local/lib/python3.10/dist-packages (from llama-cpp-python==0.2.69) (1.26.4)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2>=2.11.3->llama-cpp-python
==0.2.69) (2.1.5)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manag
er. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manag
er. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
Requirement already satisfied: rich in /usr/local/lib/python3.10/dist-packages (13.8.1)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from rich) (2.18.0)
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from rich) (3.0.0)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-packages (from markdown-it-py>=2.2.0->rich) (0.1.2)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manag
er. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (4.66.5)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manag
er. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
Requirement already satisfied: huggingface-hub in /usr/local/lib/python3.10/dist-packages (0.23.2)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub) (4.1
2.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from huggingface-hub) (3.15.4)
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub) (4.66.5)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub) (2024.6.1)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub) (6.0.2)
Requirement already satisfied: packaging>=20.9 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub) (24.1)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from huggingface-hub) (2.32.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub) (2
024.7.4)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-h
ub) (3.3.2)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub)
(2.2.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub) (3.8)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manag
er. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
Requirement already satisfied: datasets in /usr/local/lib/python3.10/dist-packages (2.14.0)
Requirement already satisfied: dill<0.3.8,>=0.3.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (0.3.7)
Requirement already satisfied: huggingface-hub<1.0.0,>=0.14.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (0.23.
2)
```

```
Requirement already satisfied: tqdm>=4.62.1 in /usr/local/lib/python3.10/dist-packages (from datasets) (4.66.5)
Requirement already satisfied: multiprocess in /usr/local/lib/python3.10/dist-packages (from datasets) (0.70.15)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from datasets) (2.2.2)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from datasets) (1.26.4)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages (from datasets) (3.10.5)
Requirement already satisfied: requests>=2.19.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (2.32.3)
Requirement already satisfied: xxhash in /usr/local/lib/python3.10/dist-packages (from datasets) (3.5.0)
Requirement already satisfied: pyarrow>=8.0.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (15.0.2)
Requirement already satisfied: fsspec[http]>=2021.11.1 in /usr/local/lib/python3.10/dist-packages (from datasets) (2024.6.1)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from datasets) (6.0.2)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from datasets) (24.1)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (6.1.0)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (2.4.
0)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.3.1)
Requirement already satisfied: async-timeout<5.0,>=4.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (4.0.
3)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.12.1)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (24.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.4.1)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0.0,>=0.14.0->dataset
s) (3.15.4)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0.
0,>=0.14.0->datasets) (4.12.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->datasets) (3.8)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->datasets)
(2024.7.4)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->datas
ets) (3.3.2)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->datasets)
(2.2.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2.9.0.
post0)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2024.1)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2024.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas->dataset
s) (1.16.0)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manag
er. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
```

In [ ]:

```
# !pip install numpy
!pip install numpy
```

```
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (1.26.4)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manag
er. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
```

1.1 (B) Import necessary libraries (2 Marks)

In [ ]:
```python
#  Import hf_hub_download from the huggingface_hub
#  Import Llama from the llama_cpp library
# Import load_dataset from datasets
# Import hf_hub_download from the huggingface_hub
from huggingface_hub import hf_hub_download

# Import Llama from the llama_cpp library
from llama_cpp import Llama

# Import load_dataset from datasets
from datasets import load_dataset
```

In [ ]:
```python
from sklearn.model_selection import train_test_split
from sklearn.metrics import f1_score
```

In [ ]:
```python
import datasets
import pandas as pd
import numpy as np
import torch
```

In [ ]:
```python
from tqdm import tqdm
```

In [ ]:
```python
import json
import re
```

1.2 Model Setup (3 Marks)

(A) Setup model name and base name (2 Marks)

(B) Create model path variable (1 Marks)

In [ ]:
```python
#  model name should be the "Llama-13B-chat-GGUF" from "TheBloke"
# model basename should be the Quantized model we want to use here - This should be the "Q5_K_M.gguf"
```

```
model_name = "TheBloke/Llama-2-13B-chat-GGUF"
model_basename = "llama-2-13b-chat.Q5_K_M.gguf" # the model is in gguf format
```

In [ ]:
```
model_path = hf_hub_download(
    repo_id=model_name, # mention the variable "model_name" here
    filename=model_basename # mention the variable "model_basename" here
    )
```

In [ ]:
```

```

If you run out of memory, use the commented command below. This issue occurs when the model is already loaded into memory.

In [ ]:
```
# del lcpp_llm
```

In [ ]:
```
lcpp_llm = Llama(
        model_path=model_path,
        n_threads=2,  # CPU cores
        n_batch=512,  # Should be between 1 and n_ctx, consider the amount of VRAM in your GPU.
        n_gpu_layers=43,  # Change this value based on your model and your GPU VRAM pool.
        n_ctx=4096,  # Context window
    )
```

```
llama_model_loader: loaded meta data with 19 key-value pairs and 363 tensors from /root/.cache/huggingface/hub/models--TheBloke-
--Llama-2-13B-chat-GGUF/snapshots/4458acc949de0a9914c3eab623904d4fe999050a/llama-2-13b-chat.Q5_K_M.gguf (version GGUF V2)
llama_model_loader: Dumping metadata keys/values. Note: KV overrides do not apply in this output.
llama_model_loader: - kv   0:                       general.architecture str              = llama
llama_model_loader: - kv   1:                               general.name str              = LLaMA v2
llama_model_loader: - kv   2:                      llama.context_length u32              = 4096
llama_model_loader: - kv   3:                    llama.embedding_length u32              = 5120
llama_model_loader: - kv   4:                         llama.block_count u32              = 40
llama_model_loader: - kv   5:                 llama.feed_forward_length u32              = 13824
llama_model_loader: - kv   6:                llama.rope.dimension_count u32              = 128
llama_model_loader: - kv   7:                llama.attention.head_count u32              = 40
llama_model_loader: - kv   8:             llama.attention.head_count_kv u32              = 40
llama_model_loader: - kv   9:    llama.attention.layer_norm_rms_epsilon f32              = 0.000010
llama_model_loader: - kv  10:                          general.file_type u32              = 17
llama_model_loader: - kv  11:                       tokenizer.ggml.model str              = llama
llama_model_loader: - kv  12:                      tokenizer.ggml.tokens arr[str,32000]   = ["<unk>", "<s>", "</s>", "<0x00>", "
<...
llama_model_loader: - kv  13:                      tokenizer.ggml.scores arr[f32,32000]   = [0.000000, 0.000000, 0.000000, 0.000
0...
llama_model_loader: - kv  14:                  tokenizer.ggml.token_type arr[i32,32000]   = [2, 3, 3, 6, 6, 6, 6, 6, 6, 6, 6, 6,
...
llama_model_loader: - kv  15:                tokenizer.ggml.bos_token_id u32              = 1
llama_model_loader: - kv  16:                tokenizer.ggml.eos_token_id u32              = 2
llama_model_loader: - kv  17:            tokenizer.ggml.unknown_token_id u32              = 0
llama_model_loader: - kv  18:               general.quantization_version u32              = 2
llama_model_loader: - type  f32:   81 tensors
llama_model_loader: - type q5_K:  241 tensors
llama_model_loader: - type q6_K:   41 tensors
llm_load_vocab: special tokens definition check successful ( 259/32000 ).
llm_load_print_meta: format           = GGUF V2
llm_load_print_meta: arch             = llama
llm_load_print_meta: vocab type       = SPM
llm_load_print_meta: n_vocab          = 32000
llm_load_print_meta: n_merges         = 0
llm_load_print_meta: n_ctx_train      = 4096
llm_load_print_meta: n_embd           = 5120
llm_load_print_meta: n_head           = 40
llm_load_print_meta: n_head_kv        = 40
llm_load_print_meta: n_layer          = 40
llm_load_print_meta: n_rot            = 128
llm_load_print_meta: n_embd_head_k    = 128
llm_load_print_meta: n_embd_head_v    = 128
llm_load_print_meta: n_gqa            = 1
```

```
llm_load_print_meta: n_embd_k_gqa     = 5120
llm_load_print_meta: n_embd_v_gqa     = 5120
llm_load_print_meta: f_norm_eps       = 0.0e+00
llm_load_print_meta: f_norm_rms_eps   = 1.0e-05
llm_load_print_meta: f_clamp_kqv      = 0.0e+00
llm_load_print_meta: f_max_alibi_bias = 0.0e+00
llm_load_print_meta: f_logit_scale    = 0.0e+00
llm_load_print_meta: n_ff             = 13824
llm_load_print_meta: n_expert         = 0
llm_load_print_meta: n_expert_used    = 0
llm_load_print_meta: causal attn      = 1
llm_load_print_meta: pooling type     = 0
llm_load_print_meta: rope type        = 0
llm_load_print_meta: rope scaling     = linear
llm_load_print_meta: freq_base_train  = 10000.0
llm_load_print_meta: freq_scale_train = 1
llm_load_print_meta: n_yarn_orig_ctx  = 4096
llm_load_print_meta: rope_finetuned   = unknown
llm_load_print_meta: ssm_d_conv       = 0
llm_load_print_meta: ssm_d_inner      = 0
llm_load_print_meta: ssm_d_state      = 0
llm_load_print_meta: ssm_dt_rank      = 0
llm_load_print_meta: model type       = 13B
llm_load_print_meta: model ftype      = Q5_K - Medium
llm_load_print_meta: model params     = 13.02 B
llm_load_print_meta: model size       = 8.60 GiB (5.67 BPW)
llm_load_print_meta: general.name     = LLaMA v2
llm_load_print_meta: BOS token        = 1 '<s>'
llm_load_print_meta: EOS token        = 2 '</s>'
llm_load_print_meta: UNK token        = 0 '<unk>'
llm_load_print_meta: LF token         = 13 '<0x0A>'
ggml_cuda_init: GGML_CUDA_FORCE_MMQ:   no
ggml_cuda_init: CUDA_USE_TENSOR_CORES: yes
ggml_cuda_init: found 1 CUDA devices:
  Device 0: Tesla T4, compute capability 7.5, VMM: yes
llm_load_tensors: ggml ctx size =    0.37 MiB
llm_load_tensors: offloading 40 repeating layers to GPU
llm_load_tensors: offloading non-repeating layers to GPU
llm_load_tensors: offloaded 41/41 layers to GPU
llm_load_tensors:        CPU buffer size =   107.42 MiB
llm_load_tensors:      CUDA0 buffer size =  8694.21 MiB
...................................................................................................
llama_new_context_with_model: n_ctx      = 4096
llama_new_context_with_model: n_batch    = 512
```

```
llama_new_context_with_model: n_ubatch   = 512
llama_new_context_with_model: flash_attn = 0
llama_new_context_with_model: freq_base  = 10000.0
llama_new_context_with_model: freq_scale = 1
llama_kv_cache_init:      CUDA0 KV buffer size =  3200.00 MiB
llama_new_context_with_model: KV self size  = 3200.00 MiB, K (f16): 1600.00 MiB, V (f16): 1600.00 MiB
llama_new_context_with_model:  CUDA_Host  output buffer size =     0.12 MiB
llama_new_context_with_model:      CUDA0 compute buffer size =   368.00 MiB
llama_new_context_with_model:  CUDA_Host compute buffer size =    18.01 MiB
llama_new_context_with_model: graph nodes  = 1286
llama_new_context_with_model: graph splits = 2
AVX = 1 | AVX_VNNI = 0 | AVX2 = 1 | AVX512 = 1 | AVX512_VBMI = 0 | AVX512_VNNI = 1 | FMA = 1 | NEON = 0 | ARM_FMA = 0 | F16C = 1
| FP16_VA = 0 | WASM_SIMD = 0 | BLAS = 1 | SSE3 = 1 | SSSE3 = 1 | VSX = 0 | MATMUL_INT8 = 0 | LLAMAFILE = 1 |
Model metadata: {'tokenizer.ggml.unknown_token_id': '0', 'tokenizer.ggml.eos_token_id': '2', 'general.architecture': 'llama', 'l
lama.context_length': '4096', 'general.name': 'LLaMA v2', 'llama.embedding_length': '5120', 'llama.feed_forward_length': '1382
4', 'llama.attention.layer_norm_rms_epsilon': '0.000010', 'llama.rope.dimension_count': '128', 'llama.attention.head_count': '4
0', 'tokenizer.ggml.bos_token_id': '1', 'llama.block_count': '40', 'llama.attention.head_count_kv': '40', 'general.quantization_
version': '2', 'tokenizer.ggml.model': 'llama', 'general.file_type': '17'}
Using fallback chat format: None
```

1.3 Data Preparation (3 Marks)

(A) Upload and read csv (1 Marks)

(B) Product Based DataFrame creation (2 Marks)

1.3 (A) Upload and read csv (1 Marks)

In [ ]:
```
sample_reviews_df = pd.read_csv("customer_reviews_dataset.csv")
```

In [ ]:
```
sample_reviews_df
```

Out[ ]:

	customer_id	customer_name	product_name	product_type	review_date	rating	review_sentiment	review_text	aspects_review	response	
0	CID041	Aisha Patel	Dell Inspiron 15	Laptop	01/01/2024	5	Positive	I bought this laptop for my son who is studyin...	{ battery : Positive , keyboard : Positive, di...	It's fantastic to hear that the laptop you pur...	lap
1	CID011	Liam Thomson	JBL Tune 500BT	Headphones	01/05/2024	1	Negative	I was very disappointed with these headphones....	{ sound : Negative , comfort : Negative , char...	I'm truly sorry to hear about your disappointi...	disa wi
2	CID034	Maria García	Mi Power Bank 3i	Power Bank	01/10/2023	4	Positive	Awesome power bank, it charges my phone very f...	{ battery : Positive , charging : Positive }	Thank you for your positive review of our powe...	pov
3	CID032	Jamal Johnson	Samsung Galaxy S21	Smartphone	03/03/2023	2	Negative	I bought this phone mainly for its much-hyped ...	{ display : Positive , camera : Negative , Bat...	I'm truly sorry to hear that the camera perfor...	disa wi
4	CID051	Emily Nguyen	JBL Tune 500BT	Headphones	05/25/2023	5	Positive	I love these headphones, they are amazing. The...	{ sound : Positive , comfort : Positive , char...	Thank you for your wonderful feedback on our h...	hea
5	CID012	Rafael Fernandez	Mi Power Bank 3i	Power Bank	01/13/2024	1	Negative	This is a very bad power bank, it does not cha...	{ battery : Negative , charging : Negative }	I'm deeply concerned to hear about your experi...	Tl rev
6	CID043	Sofia Chen	Dell Inspiron 15	Laptop	05/20/2023	1	Negative	I bought this laptop a month ago, and it's alr...	{ battery : Negative , keyboard : Negative , d...	I'm sorry to hear about your laptop issues and...	Tl re
7	CID063	Michael Lee	Dell Inspiron 15	Laptop	02/09/2023	2	Negative	This is a below average laptop. It has inconsi...	{ battery : Negative , keyboard : Negative , d...	I'm sorry to hear that your laptop isn't meeti...	Tl be

	customer_id	customer_name	product_name	product_type	review_date	rating	review_sentiment	review_text	aspects_review	response	
8	CID029	Olivia Rodriguez	JBL Tune 500BT	Headphones	01/02/2024	1	Negative	I bought these headphones a week ago, and they...	{ sound : Negative , comfort : Negative , char...	I apologize for the issues you're experiencing...	Th I h
9	CID080	Karan Singh	JBL Tune 500BT	Headphones	02/22/2023	5	Positive	These are the best headphones I have ever used...	{ sound : Positive , comfort : Positive , char...	Thank you for sharing your positive experience...	Th hea
10	CID007	David Kim	Dell Inspiron 15	Laptop	05/03/2023	1	Negative	I hate this laptop, it is a nightmare. I bough...	{ battery : Negative , keyboard : Negative , d...	I'm truly sorry to hear about the difficulties...	Th p
11	CID040	Reena Gupta	Mi Power Bank 3i	Power Bank	10/02/2023	5	Positive	Recently tried a power bank sharing service - ...	{ battery : Positive , charging : Negative }	Thank you for your feedback on the power bank ...	Th sh
12	CID070	Emre Çalhanoğlu	Samsung Galaxy S21	Smartphone	11/03/2023	2	Negative	I am very unhappy with this phone, it has a ve...	{ display : Negative , camera : Negative , Bat...	I'm truly sorry to hear about your dissatisfac...	Th di
13	CID060	Ketan Kopikar	JBL Tune 500BT	Headphones	01/06/2024	2	Negative	I bought these headphones just 2 weeks ago, an...	{ sound : Negative , comfort : Negative , char...	I'm sorry to hear about your negative experien...	The u he
14	CID004	Bradley Wiggins	Dell Inspiron 15	Laptop	05/10/2024	5	Positive	This is a great laptop, I am very happy with i...	{ battery : Positive , keyboard : Positive , d...	Thank you for your positive feedback on our la...	The s the
15	CID100	Milad Hosseini	Mi Power Bank 3i	Power Bank	07/17/2023	4	Positive	This is a good power bank, it does what it say...	{ battery : Positive , charging : Positive }	Thank you for sharing your positive experience...	Th pov

	customer_id	customer_name	product_name	product_type	review_date	rating	review_sentiment	review_text	aspects_review	response	
16	CID030	Rachel Smith	Samsung Galaxy S21	Smartphone	08/28/2023	4	Positive	Good phone, I am satisfied with it. The phone ...	{ display : Positive , camera : Positive , Bat...	Thank you for your positive review of the phon...	The s
17	CID010	Miora Fimanantsoa	JBL Tune 500BT	Headphones	12/12/2023	4	Positive	The headphones work well. The sound quality is...	{ sound : Positive , comfort : Positive , char...	Thank you for your review of our headphones. I...	T hea
18	CID078	Nisha Sharma	Dell Inspiron 15	Laptop	06/12/2023	2	Negative	I bought this laptop quite recently, and I alr...	{ battery : Positive , keyboard : Positive, di...	I'm sorry to hear about the issues you're expe...	The u t
19	CID087	Zulmi Virmansyah	Samsung Galaxy S21	Smartphone	05/27/2023	2	Negative	Bought this phone because I wanted to upgrade ...	{ display : Negative , camera : Positive , Bat...	I'm sorry to hear about your disappointment wi...	T upg
20	CID027	Kim Jung	Samsung Galaxy S21	Smartphone	09/28/2023	5	Positive	Amazing phone. It has a beautiful design and a...	{ display : Negative , camera : Positive , Bat...	Thank you for your enthusiastic review of the ...	T pho
21	CID085	Manuela Marin	JBL Tune 500BT	Headphones	10/01/2023	1	Negative	I was very excited to get these headphones as ...	{ sound : Negative , comfort : Negative , char...	I'm truly sorry to hear about your disappointi...	T c
22	CID076	Amit Shah	Mi Power Bank 3i	Power Bank	03/02/2023	1	Negative	This is a very poor power bank, it does not wo...	{ battery : Negative , charging : Negative }	I'm sorry to hear about your dissatisfaction w...	The
23	CID075	Jack Ryder	JBL Tune 500BT	Headphones	05/05/2023	5	Positive	Great for listening to music or podcasts. Good...	{ sound : Positive , comfort : Positive , char...	Thank you for your positive review of our head...	hea

	customer_id	customer_name	product_name	product_type	review_date	rating	review_sentiment	review_text	aspects_review	response	
24	CID046	Harvey Stark	Mi Power Bank 3i	Power Bank	07/19/2023	5	Positive	I love this power bank - it's amazing. It can ...	{ battery : Positive , charging : Positive }	Thank you for your enthusiastic feedback on ou...	Tl p
25	CID013	Ollie Stone	JBL Tune 500BT	Headphones	12/09/2023	1	Negative	I bought these headphones as a wireless option...	{ sound : Negative , comfort : Negative , char...	I'm sorry to hear about your unsatisfactory ex...	Tl re h
26	CID067	Adewale Akinfenwa	Dell Inspiron 15	Laptop	01/09/2024	4	Positive	The laptop is good enough for the price. It ha...	{ battery : Positive , keyboard : Positive , d...	Thank you for sharing your thoughts on our lap...	Tl pe
27	CID053	Pooja Jain	Dell Inspiron 15	Laptop	12/31/2023	4	Positive	Originally bought it for my work, quite happy ...	{ battery : Positive , keyboard : Positive , d...	Thank you for sharing your positive experience...	The s
28	CID082	Laura Wolvaardt	Mi Power Bank 3i	Power Bank	04/13/2023	4	Positive	Good backup for my phone & other devices. Smal...	{ battery : Positive , charging : Positive }	Thank you for your positive feedback on our po...	Tl li
29	CID099	Jim Kowalski	Dell Inspiron 15	Laptop	08/03/2023	1	Negative	I thought this would be the laptop I could use...	{ battery : Negative , keyboard : Negative , d...	I'm sorry to hear about the issues you're expe...	Tl p

1.3 (B) Product Based DataFrame creation (2 Marks)

In [ ]:
```
# Create a "Laptop" reviews DaatFrame based on the "product_type" column in the dataset
laptop_reviews = sample_reviews_df[sample_reviews_df['product_type'] == 'Laptop']

# Create a "Headphones" reviews DaatFrame based on the "product_type" column in the dataset
headphones_reviews =sample_reviews_df[sample_reviews_df['product_type'] == 'Headphones']

# Create a "Smartphone" reviews DaatFrame based on the "product_type" column in the dataset
```

```
smartphone_reviews = sample_reviews_df[sample_reviews_df['product_type'] == 'Smartphone']

# Create a "Power Bank" reviews DaatFrame based on the "product_type" column in the dataset
power_bank_reviews = sample_reviews_df[sample_reviews_df['product_type'] == 'Power Bank']
```

In [ ]:
```
laptop_gold_examples = laptop_reviews.sample(2, random_state=40)
headphones_gold_examples = headphones_reviews.sample(2, random_state=40)
smartphone_gold_examples = smartphone_reviews.sample(2, random_state=40)
power_bank_gold_examples = power_bank_reviews.sample(2, random_state=40)

# Concatenate positive and negative gold examples
sample_reviews_gold_examples_df = pd.concat([laptop_gold_examples, headphones_gold_examples,smartphone_gold_examples,power_bank_g

# Create the training set by excluding gold examples
sample_reviews_examples_df = sample_reviews_df.drop(index=sample_reviews_gold_examples_df.index)

# Convert gold examples to JSON
columns_to_select = ['review_text', 'product_type','aspects_review']
gold_examples_json = sample_reviews_gold_examples_df[columns_to_select].to_json(orient='records')

# Print the first record from the JSON
print(json.loads(gold_examples_json)[0])

# Print the shapes of the datasets
print("Training Set Shape:", sample_reviews_examples_df.shape)
print("Gold Examples Shape:", sample_reviews_gold_examples_df.shape)
```

```
{'review_text': "Originally bought it for my work, quite happy with it so far! Fast, reliable, easy to use and has a good webca
m. Display is good and battery backup is also great. The keyboard is a joy to type on, gives me the old typewriter vibes! Quickl
y become my main laptop for everyday use, and I'm very satisfied with my purchase.", 'product_type': 'Laptop', 'aspects_review':
'{ battery : Positive , keyboard : Positive , display : Positive }'}
Training Set Shape: (22, 11)
Gold Examples Shape: (8, 11)
```

In [ ]:
```
sample_reviews_gold_examples_df
```

Out[ ]:

	customer_id	customer_name	product_name	product_type	review_date	rating	review_sentiment	review_text	aspects_review	response	sur
27	CID053	Pooja Jain	Dell Inspiron 15	Laptop	12/31/2023	4	Positive	Originally bought it for my work, quite happy ...	{ battery : Positive , keyboard : Positive , d...	Thank you for sharing your positive experience...	The custo satisfie the
14	CID004	Bradley Wiggins	Dell Inspiron 15	Laptop	05/10/2024	5	Positive	This is a great laptop, I am very happy with i...	{ battery : Positive , keyboard : Positive , d...	Thank you for your positive feedback on our la...	The custo satisfie their lapt
23	CID075	Jack Ryder	JBL Tune 500BT	Headphones	05/05/2023	5	Positive	Great for listening to music or podcasts. Good...	{ sound : Positive , comfort : Positive , char...	Thank you for your positive review of our head...	Cus headpho good
13	CID060	Ketan Kopikar	JBL Tune 500BT	Headphones	01/06/2024	2	Negative	I bought these headphones just 2 weeks ago, an...	{ sound : Negative , comfort : Negative , char...	I'm sorry to hear about your negative experien...	The custo unhapp headph
16	CID030	Rachel Smith	Samsung Galaxy S21	Smartphone	08/28/2023	4	Positive	Good phone, I am satisfied with it. The phone ...	{ display : Positive , camera : Positive , Bat...	Thank you for your positive review of the phon...	The custo satisfie their
3	CID032	Jamal Johnson	Samsung Galaxy S21	Smartphone	03/03/2023	2	Negative	I bought this phone mainly for its much-hyped ...	{ display : Positive , camera : Negative , Bat...	I'm truly sorry to hear that the camera perfor...	Th exp disappoi with the
24	CID046	Harvey Stark	Mi Power Bank 3i	Power Bank	07/19/2023	5	Positive	I love this power bank - it's amazing. It can ...	{ battery : Positive , charging : Positive }	Thank you for your enthusiastic feedback on ou...	The cus prais power cap
5	CID012	Rafael Fernandez	Mi Power Bank 3i	Power Bank	01/13/2024	1	Negative	This is a very bad	{ battery : Negative ,	I'm deeply concerned	The cus reviews

customer_id	customer_name	product_name	product_type	review_date	rating	review_sentiment	review_text	aspects_review	response	sur
							power bank, it does not cha...	charging : Negative }	to hear about your experi...	powe

In [ ]:
```
gold_examples = (
        sample_reviews_gold_examples_df.loc[:, columns_to_select]
                                       .sample(8, random_state=40) #<- ensures that gold examples are the same for every session
                                       .to_json(orient='records')
)
```

1.4 Derive Prompt (7 Marks)

1.4 (A) Create a Few Shot System Message (5 Marks)

1.4 (B) Combine user_prompt and system_message to create the prompt (2 Marks)

The functions that we used to assemble examples, create prompts and to evaluate these prompts are broadly the same as in the case of sentiment analysis. However, we should adapt the few-shot examples to now account for the multiple aspects.

1.4 (A) Create a Few Shot System Message (5 Marks)

In [ ]:
```
few_shot_system_message = """
You are a product review classification assistant. Your task is to classify the product type and identify the sentiment towards k
output should be in json format
Here are a few examples of how you should classify the reviews:

Example 1:
Review: "This laptop's performance is excellent, but the battery life is disappointing."
Product Type: Laptop
Aspects:
- Performance: Positive
- Battery: Negative

Example 2:
Review: "The headphones have great sound quality, but they are uncomfortable to wear for long periods."
Product Type: Headphones
Aspects:
- Sound Quality: Positive
```

```
- Comfort: Negative

Example 3:
Review: "The smartphone has a fantastic camera, but it overheats during extended use."
Product Type: Smartphone
Aspects:
- Camera: Positive
- Overheating: Negative

Please follow this format when classifying reviews, identifying both the product type and relevant aspects.
"""
```

In [ ]:
```
system_message = """
You are a product review classification assistant. Your task is to classify the product type and identify the sentiment towards k
output should be in json format
"""
few_shot_examples = """
Example 1: "Great camera but bad battery." -> Product: Smartphone, Aspects_review: Camera: Positive, Battery: Negative
Example 2: "Good sound but uncomfortable fit." -> Product: Headphones, Aspects_review: Sound: Positive, Comfort: Negative
"""
new_review = "example is user The laptop is fast, but the screen quality is poor.new_review will be product wise review"
temp = 0.7
```

1.4 (B) Combine user_prompt and system_message to create the prompt (2 Marks)

In [ ]:
```
def generate_llama_response( system_message ,  few_shot_examples  , new_review , temp ):

    # Combine user_prompt and system_message to create the prompt
    prompt = f"[INST]{system_message}\n{few_shot_examples}\nUser: ```{user_message_template.format(review=new_review)}```[INST]"

    # Generate a response from the LLM model
    response = lcpp_llm(
        prompt=prompt,
        max_tokens=256,
        temperature=temp,
        top_p=0.95,
        repeat_penalty=1.2,
        top_k=50,
        stop=['INST'],
        echo=False
    )

    # Extract and return the response text
```

```
    response_text = response["choices"][0]["text"]
    return response_text
```

In [ ]:

In [ ]:
```
def create_examples_with_seed(dataset, n=2, random_seed=None):
    """
    Return two DataFrames with randomized examples of size 2n with two classes.
    Create subsets of each class, choose random samples from the subsets,
    merge and randomize the order of samples in the merged list.
    Each run of this function creates a different random sample of examples
    chosen from the training data.

    Args:
        dataset (DataFrame): A DataFrame with examples (text + label)
        n (int): number of examples of each class to be selected
        random_seed (int): seed for reproducibility (default is None)

    Output:
        few_shot_examples_df (DataFrame): A DataFrame with examples in random order
        new_df (DataFrame): A new DataFrame excluding selected examples
    """

    laptop_reviews = (dataset.product_type == 'Laptop')
    headphone_reviews = (dataset.product_type == 'Headphones')
    power_bank_reviews = (dataset.product_type == 'Power Bank')
    smartphone_reviews = (dataset.product_type == 'Smartphone')
    columns_to_select = ['review_text', 'product_type','aspects_review']

    # Set a fixed random seed for reproducibility
    np.random.seed(random_seed)

    laptop_examples = dataset.loc[laptop_reviews, columns_to_select].sample(n)
    headphone_examples = dataset.loc[headphone_reviews, columns_to_select].sample(n)
    power_bank_examples = dataset.loc[power_bank_reviews, columns_to_select].sample(n)
    smartphone_examples = dataset.loc[smartphone_reviews, columns_to_select].sample(n)

    few_shot_examples_df = pd.concat([laptop_examples, headphone_examples, power_bank_examples, smartphone_examples])

    # sampling without replacement is equivalent to random shuffling
    few_shot_examples_df = few_shot_examples_df.sample( 4*n, replace=False)

    # Create a new DataFrame excluding selected examples
```

```
    new_df = dataset.drop(index=few_shot_examples_df.index)

    return few_shot_examples_df, new_df
```

In [ ]:

```
def compute_combined_aspect_and_product_accuracy(df):
    correct_predictions_count = 0

    # Function to parse aspect-based sentiment string into a dictionary
    def parse_aspects(aspect_string):
        aspect_string= str(aspect_string)
        aspect_string = re.sub(r'[{}]', '', aspect_string)  # Remove curly braces
        aspects = re.split(r',\s*', aspect_string)  # Split into individual aspects

        try:
         dict_res = dict(re.split(r'\s*:\s*', aspect) for aspect in aspects if aspect)
        except:
          dict_res={}

        return dict_res


    # Function to normalize text by removing spaces and converting to lowercase
    def normalize_text(text):
        return ''.join(text.split()).lower() if text is not None else None

    # Iterate over each row to compare product type and aspect-based sentiments
    for index, row in df.iterrows():
        predicted_product_type = normalize_text(row['predicted_product_type'])
        actual_product_type = normalize_text(row['product_type'])
        predicted_aspects = parse_aspects(row['predicted_aspect_based_sentiment'])
        actual_aspects = parse_aspects(row['aspects_review'])

        # Check if product type matches and all aspects match in sentiment
        if predicted_product_type == actual_product_type and \
          predicted_aspects != '' and \
          all(predicted_aspects.get(key, '').lower() == value.lower() for key, value in actual_aspects.items()):
            correct_predictions_count += 1

    # Calculate accuracy
    accuracy = (correct_predictions_count / len(df)) * 100
    return accuracy
```

1.5 Measuring prompt performance (7 Marks)

1.5 (A) Generate response and compute accuracy (5 Marks)

1.5 (B) Evaluate the results (2 marks)

In [ ]:
```
user_message_template = "{review}"
```

In [ ]:
```
def extract_product_type(text):
    text=str(text)
    match = re.search(r'\[([^:]+)', text)
    return match.group(1).strip() if match else None

def extract_aspect_based_sentiment(text):
    text=str(text)
    match = re.search(r'\{([^}]+)\}', text)
    return "{ " + match.group(1).strip() + " }" if match else None
```

In [ ]:
```
accuracy_list = []
```

In [ ]:
```
sample_reviews_examples_df
```

Out[ ]:

	customer_id	customer_name	product_name	product_type	review_date	rating	review_sentiment	review_text	aspects_review	response	
0	CID041	Aisha Patel	Dell Inspiron 15	Laptop	01/01/2024	5	Positive	I bought this laptop for my son who is studyin...	{ battery : Positive , keyboard : Positive, di...	It's fantastic to hear that the laptop you pur...	lap
1	CID011	Liam Thomson	JBL Tune 500BT	Headphones	01/05/2024	1	Negative	I was very disappointed with these headphones....	{ sound : Negative , comfort : Negative , char...	I'm truly sorry to hear about your disappointi...	disa wi
2	CID034	Maria García	Mi Power Bank 3i	Power Bank	01/10/2023	4	Positive	Awesome power bank, it charges my phone very f...	{ battery : Positive , charging : Positive }	Thank you for your positive review of our powe...	pov
4	CID051	Emily Nguyen	JBL Tune 500BT	Headphones	05/25/2023	5	Positive	I love these headphones, they are amazing. The...	{ sound : Positive , comfort : Positive , char...	Thank you for your wonderful feedback on our h...	hea
6	CID043	Sofia Chen	Dell Inspiron 15	Laptop	05/20/2023	1	Negative	I bought this laptop a month ago, and it's alr...	{ battery : Negative , keyboard : Negative , d...	I'm sorry to hear about your laptop issues and...	Tl re
7	CID063	Michael Lee	Dell Inspiron 15	Laptop	02/09/2023	2	Negative	This is a below average laptop. It has inconsi...	{ battery : Negative , keyboard : Negative , d...	I'm sorry to hear that your laptop isn't meeti...	Tl be
8	CID029	Olivia Rodriguez	JBL Tune 500BT	Headphones	01/02/2024	1	Negative	I bought these headphones a week ago, and they...	{ sound : Negative , comfort : Negative , char...	I apologize for the issues you're experiencing...	Tl h
9	CID080	Karan Singh	JBL Tune 500BT	Headphones	02/22/2023	5	Positive	These are the best headphones I have ever used...	{ sound : Positive , comfort : Positive , char...	Thank you for sharing your positive experience...	Tl hea

	customer_id	customer_name	product_name	product_type	review_date	rating	review_sentiment	review_text	aspects_review	response	
10	CID007	David Kim	Dell Inspiron 15	Laptop	05/03/2023	1	Negative	I hate this laptop, it is a nightmare. I bough...	{ battery : Negative , keyboard : Negative , d...	I'm truly sorry to hear about the difficulties...	Tl p
11	CID040	Reena Gupta	Mi Power Bank 3i	Power Bank	10/02/2023	5	Positive	Recently tried a power bank sharing service - ...	{ battery : Positive , charging : Negative }	Thank you for your feedback on the power bank ...	Tl sl
12	CID070	Emre Çalhanoğlu	Samsung Galaxy S21	Smartphone	11/03/2023	2	Negative	I am very unhappy with this phone, it has a ve...	{ display : Negative , camera : Negative , Bat...	I'm truly sorry to hear about your dissatisfac...	Tl di
15	CID100	Milad Hosseini	Mi Power Bank 3i	Power Bank	07/17/2023	4	Positive	This is a good power bank, it does what it say...	{ battery : Positive , charging : Positive }	Thank you for sharing your positive experience...	Tl pov
17	CID010	Miora Fimanantsoa	JBL Tune 500BT	Headphones	12/12/2023	4	Positive	The headphones work well. The sound quality is...	{ sound : Positive , comfort : Positive , char...	Thank you for your review of our headphones. I...	Tl hea
18	CID078	Nisha Sharma	Dell Inspiron 15	Laptop	06/12/2023	2	Negative	I bought this laptop quite recently, and I alr...	{ battery : Positive , keyboard : Positive, di...	I'm sorry to hear about the issues you're expe...	The u t
19	CID087	Zulmi Virmansyah	Samsung Galaxy S21	Smartphone	05/27/2023	2	Negative	Bought this phone because I wanted to upgrade ...	{ display : Negative , camera : Positive , Bat...	I'm sorry to hear about your disappointment wi...	Tl upg
20	CID027	Kim Jung	Samsung Galaxy S21	Smartphone	09/28/2023	5	Positive	Amazing phone. It has a beautiful design and a...	{ display : Negative , camera : Positive , Bat...	Thank you for your enthusiastic review of the ...	Tl pho
21	CID085	Manuela Marin	JBL Tune 500BT	Headphones	10/01/2023	1	Negative	I was very excited to get	{ sound : Negative ,	I'm truly sorry to hear about	Tl

	customer_id	customer_name	product_name	product_type	review_date	rating	review_sentiment	review_text	aspects_review	response	
								these headphones as ...	comfort : Negative , char...	your disappointi...	c
22	CID076	Amit Shah	Mi Power Bank 3i	Power Bank	03/02/2023	1	Negative	This is a very poor power bank, it does not wo...	{ battery : Negative , charging : Negative }	I'm sorry to hear about your dissatisfaction w...	The
25	CID013	Ollie Stone	JBL Tune 500BT	Headphones	12/09/2023	1	Negative	I bought these headphones as a wireless option...	{ sound : Negative , comfort : Negative , char...	I'm sorry to hear about your unsatisfactory ex...	Tl re h
26	CID067	Adewale Akinfenwa	Dell Inspiron 15	Laptop	01/09/2024	4	Positive	The laptop is good enough for the price. It ha...	{ battery : Positive , keyboard : Positive , d...	Thank you for sharing your thoughts on our lap...	Tl pe
28	CID082	Laura Wolvaardt	Mi Power Bank 3i	Power Bank	04/13/2023	4	Positive	Good backup for my phone & other devices. Smal...	{ battery : Positive , charging : Positive }	Thank you for your positive feedback on our po...	Tl li
29	CID099	Jim Kowalski	Dell Inspiron 15	Laptop	08/03/2023	1	Negative	I thought this would be the laptop I could use...	{ battery : Negative , keyboard : Negative , d...	I'm sorry to hear about the issues you're expe...	Tl p

1.5 (A) Generate response and compute accuracy (5 Marks)

In [ ]:
```
for i in range(3):
    few_shot_examples_df ,sample_reviews_df = create_examples_with_seed(sample_reviews_examples_df, n=1 , random_seed = i)

    review_example1 = few_shot_examples_df.iloc[0].review_text
    review_example2 = few_shot_examples_df.iloc[1].review_text
    review_example3 = few_shot_examples_df.iloc[2].review_text
    review_example4 = few_shot_examples_df.iloc[3].review_text
```

```
assistant_output_example1 = "[ " + few_shot_examples_df.iloc[0].product_type.lower() + " : " + few_shot_examples_df.iloc[0].a
assistant_output_example2 = "[ " + few_shot_examples_df.iloc[1].product_type.lower() + " : " + few_shot_examples_df.iloc[1].a
assistant_output_example3 = "[ " + few_shot_examples_df.iloc[2].product_type.lower() + " : " + few_shot_examples_df.iloc[2].a
assistant_output_example4 = "[ " + few_shot_examples_df.iloc[3].product_type.lower() + " : " + few_shot_examples_df.iloc[3].a

few_shot_examples = [
    {'role':'user', 'content': user_message_template.format(review=review_example1)},
    {'role':'assistant', 'content': f"{assistant_output_example1}"},
    {'role':'user', 'content': user_message_template.format(review=review_example2)},
    {'role':'assistant', 'content': f"{assistant_output_example2}"},
    {'role':'user', 'content': user_message_template.format(review=review_example3)},
    {'role':'assistant', 'content': f"{assistant_output_example3}"},
    {'role':'user', 'content': user_message_template.format(review=review_example4)},
    {'role':'assistant', 'content': f"{assistant_output_example4}"}
                    ]

few_shot_examples_str = json.dumps(few_shot_examples)


sample_reviews = sample_reviews_df.review_text.values
sentiment_predictions = []
# generate_llama_response( few_shot_system_message , few_shot_examples_str  , input_text , 0.1 )
for sample_review in tqdm(sample_reviews):
    try:
        #predict sentiments (2 Marks)
        sentiment_predictions.append(generate_llama_response( few_shot_system_message ,few_shot_examples_str , sample_review
    except Exception as e:
        print(e)
        sentiment_predictions.append("")
sentiment_predictions
sample_reviews_df["sentiment_prediction"] = sentiment_predictions

pattern = r'\[\s*.*?\s*:\s*\{.*?\}\s*\]'

# Function to extract the sentiment data
def extract_sentiment(text):
    match = re.findall(pattern, text)
    return match[0] if match else None


# Apply the function to each row in the DataFrame
sample_reviews_df['extracted_sentiment'] = sample_reviews_df['sentiment_prediction'].apply(extract_sentiment)

sample_reviews_df
```

```python
# Function to extract product type
print(sample_reviews_df['extracted_sentiment'])

# Apply the functions to each row in the DataFrame
try:
        sample_reviews_df['predicted_product_type'] = sample_reviews_df['extracted_sentiment'].apply(extract_product_type)
        sample_reviews_df['predicted_aspect_based_sentiment'] = sample_reviews_df['extracted_sentiment'].apply(extract_aspect
except Exception as e:
        print(e)
        print(sample_reviews_df['predicted_product_type'] )

sample_reviews_df.predicted_aspect_based_sentiment.value_counts()



# Compute Combimed Accuracy (1 Mark) - Call the "compute_combined_aspect_and_product_accuracy" with "sample_reviews_df" as in
combined_accuracy = compute_combined_aspect_and_product_accuracy(sample_reviews_df)

print(f"Combined Product Type and Aspect-Based Sentiment Accuracy: {combined_accuracy}%")
res = combined_accuracy
# Append the "res" to the "accuracy_list" (1 Mark)
accuracy_list.append(res)
```

```
  0%|          | 0/18 [00:00<?, ?it/s]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     137.78 ms /   256 runs   (    0.54 ms per token,  1858.06 tokens per second)
llama_print_timings: prompt eval time =    1433.63 ms /   729 tokens (    1.97 ms per token,   508.50 tokens per second)
llama_print_timings:        eval time =   12565.53 ms /   255 runs   (   49.28 ms per token,    20.29 tokens per second)
llama_print_timings:       total time =   14898.49 ms /   984 tokens
  6%|▌         | 1/18 [00:14<04:13, 14.91s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =      94.05 ms /   177 runs   (    0.53 ms per token,  1882.08 tokens per second)
llama_print_timings: prompt eval time =     380.92 ms /   101 tokens (    3.77 ms per token,   265.15 tokens per second)
llama_print_timings:        eval time =    8746.62 ms /   176 runs   (   49.70 ms per token,    20.12 tokens per second)
llama_print_timings:       total time =    9742.30 ms /   277 tokens
 11%|█         | 2/18 [00:24<03:09, 11.87s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     135.42 ms /   256 runs   (    0.53 ms per token,  1890.40 tokens per second)
llama_print_timings: prompt eval time =     375.78 ms /    92 tokens (    4.08 ms per token,   244.82 tokens per second)
llama_print_timings:        eval time =   12697.32 ms /   255 runs   (   49.79 ms per token,    20.08 tokens per second)
llama_print_timings:       total time =   13980.87 ms /   347 tokens
 17%|█▋        | 3/18 [00:38<03:12, 12.84s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     135.82 ms /   256 runs   (    0.53 ms per token,  1884.82 tokens per second)
llama_print_timings: prompt eval time =     496.66 ms /   175 tokens (    2.84 ms per token,   352.35 tokens per second)
llama_print_timings:        eval time =   12830.76 ms /   255 runs   (   50.32 ms per token,    19.87 tokens per second)
llama_print_timings:       total time =   14237.61 ms /   430 tokens
 22%|██▏       | 4/18 [00:52<03:07, 13.39s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     133.02 ms /   256 runs   (    0.52 ms per token,  1924.48 tokens per second)
llama_print_timings: prompt eval time =     380.66 ms /    99 tokens (    3.85 ms per token,   260.08 tokens per second)
llama_print_timings:        eval time =   12789.96 ms /   255 runs   (   50.16 ms per token,    19.94 tokens per second)
llama_print_timings:       total time =   14085.65 ms /   354 tokens
 28%|██▊       | 5/18 [01:06<02:57, 13.65s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     133.45 ms /   256 runs   (    0.52 ms per token,  1918.29 tokens per second)
llama_print_timings: prompt eval time =     407.55 ms /   123 tokens (    3.31 ms per token,   301.81 tokens per second)
llama_print_timings:        eval time =   12812.08 ms /   255 runs   (   50.24 ms per token,    19.90 tokens per second)
llama_print_timings:       total time =   14143.46 ms /   378 tokens
 33%|███▎      | 6/18 [01:21<02:45, 13.82s/it]Llama.generate: prefix-match hit
```

```
llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =      99.89 ms /   187 runs   (    0.53 ms per token,  1872.12 tokens per second)
llama_print_timings: prompt eval time =     468.35 ms /   134 tokens (    3.50 ms per token,   286.11 tokens per second)
llama_print_timings:        eval time =    9322.50 ms /   186 runs   (   50.12 ms per token,    19.95 tokens per second)
llama_print_timings:       total time =   10454.87 ms /   320 tokens
 39%|███▉      | 7/18 [01:31<02:19, 12.72s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     132.59 ms /   256 runs   (    0.52 ms per token,  1930.75 tokens per second)
llama_print_timings: prompt eval time =     465.02 ms /   131 tokens (    3.55 ms per token,   281.71 tokens per second)
llama_print_timings:        eval time =   12813.43 ms /   255 runs   (   50.25 ms per token,    19.90 tokens per second)
llama_print_timings:       total time =   14198.44 ms /   386 tokens
 44%|████▍     | 8/18 [01:45<02:11, 13.19s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     137.14 ms /   256 runs   (    0.54 ms per token,  1866.75 tokens per second)
llama_print_timings: prompt eval time =     366.14 ms /    70 tokens (    5.23 ms per token,   191.19 tokens per second)
llama_print_timings:        eval time =   12741.14 ms /   255 runs   (   49.97 ms per token,    20.01 tokens per second)
llama_print_timings:       total time =   14037.53 ms /   325 tokens
 50%|████▉     | 9/18 [01:59<02:01, 13.46s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     135.74 ms /   256 runs   (    0.53 ms per token,  1885.94 tokens per second)
llama_print_timings: prompt eval time =     368.30 ms /    73 tokens (    5.05 ms per token,   198.21 tokens per second)
llama_print_timings:        eval time =   12762.79 ms /   255 runs   (   50.05 ms per token,    19.98 tokens per second)
llama_print_timings:       total time =   14057.77 ms /   328 tokens
 56%|█████▌    | 10/18 [02:13<01:49, 13.65s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     137.70 ms /   256 runs   (    0.54 ms per token,  1859.14 tokens per second)
llama_print_timings: prompt eval time =     388.89 ms /   112 tokens (    3.47 ms per token,   288.00 tokens per second)
llama_print_timings:        eval time =   12810.99 ms /   255 runs   (   50.24 ms per token,    19.90 tokens per second)
llama_print_timings:       total time =   14132.72 ms /   367 tokens
 61%|██████    | 11/18 [02:28<01:36, 13.80s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     135.92 ms /   256 runs   (    0.53 ms per token,  1883.43 tokens per second)
llama_print_timings: prompt eval time =     387.48 ms /   109 tokens (    3.55 ms per token,   281.30 tokens per second)
llama_print_timings:        eval time =   12811.56 ms /   255 runs   (   50.24 ms per token,    19.90 tokens per second)
llama_print_timings:       total time =   14137.53 ms /   364 tokens
 67%|██████▋   | 12/18 [02:42<01:23, 13.90s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
```

```
llama_print_timings:      sample time =     112.06 ms /   210 runs   (    0.53 ms per token,  1874.03 tokens per second)
llama_print_timings: prompt eval time =     405.51 ms /   121 tokens (    3.35 ms per token,   298.39 tokens per second)
llama_print_timings:        eval time =   10514.06 ms /   209 runs   (   50.31 ms per token,    19.88 tokens per second)
llama_print_timings:       total time =   11675.96 ms /   330 tokens
 72%|███████▏    | 13/18 [02:53<01:06, 13.23s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     134.78 ms /   256 runs   (    0.53 ms per token,  1899.36 tokens per second)
llama_print_timings: prompt eval time =     374.88 ms /    89 tokens (    4.21 ms per token,   237.41 tokens per second)
llama_print_timings:        eval time =   12797.00 ms /   255 runs   (   50.18 ms per token,    19.93 tokens per second)
llama_print_timings:       total time =   14103.76 ms /   344 tokens
 78%|███████▊    | 14/18 [03:07<00:53, 13.50s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     134.19 ms /   256 runs   (    0.52 ms per token,  1907.81 tokens per second)
llama_print_timings: prompt eval time =     482.15 ms /   152 tokens (    3.17 ms per token,   315.26 tokens per second)
llama_print_timings:        eval time =   12865.08 ms /   255 runs   (   50.45 ms per token,    19.82 tokens per second)
llama_print_timings:       total time =   14279.58 ms /   407 tokens
 83%|████████▎   | 15/18 [03:22<00:41, 13.74s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     123.38 ms /   229 runs   (    0.54 ms per token,  1856.11 tokens per second)
llama_print_timings: prompt eval time =     378.05 ms /    94 tokens (    4.02 ms per token,   248.64 tokens per second)
llama_print_timings:        eval time =   11441.37 ms /   228 runs   (   50.18 ms per token,    19.93 tokens per second)
llama_print_timings:       total time =   12654.17 ms /   322 tokens
 89%|████████▉   | 16/18 [03:34<00:26, 13.41s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     133.96 ms /   256 runs   (    0.52 ms per token,  1911.03 tokens per second)
llama_print_timings: prompt eval time =     373.17 ms /    80 tokens (    4.66 ms per token,   214.38 tokens per second)
llama_print_timings:        eval time =   12789.62 ms /   255 runs   (   50.16 ms per token,    19.94 tokens per second)
llama_print_timings:       total time =   14097.09 ms /   335 tokens
 94%|█████████▍  | 17/18 [03:49<00:13, 13.62s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     135.89 ms /   256 runs   (    0.53 ms per token,  1883.88 tokens per second)
llama_print_timings: prompt eval time =     368.17 ms /    73 tokens (    5.04 ms per token,   198.28 tokens per second)
llama_print_timings:        eval time =   12779.23 ms /   255 runs   (   50.11 ms per token,    19.95 tokens per second)
llama_print_timings:       total time =   14086.72 ms /   328 tokens
100%|██████████| 18/18 [04:03<00:00, 13.51s/it]
```

```
0                                                    None
1                                                    None
2                                                    None
6                                                    None
7                                                    None
8     [ headphones : { battery : negative , comfort ...
9                                                    None
10                                                   None
11                                                   None
15                                                   None
17                                                   None
18                                                   None
19    [ smartphone : { battery : negative , camera :...
20    [ headphones : { sound : positive , comfort : ...
21                                                   None
25    [ headphones : { comfort : negative , stabilit...
26    [ headphones : { sound : positive , comfort : ...
28                                                   None
Name: extracted_sentiment, dtype: object
Combined Product Type and Aspect-Based Sentiment Accuracy: 0.0%
```

```
  0%|          | 0/18 [00:00<?, ?it/s]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     120.71 ms /   231 runs   (    0.52 ms per token,  1913.66 tokens per second)
llama_print_timings: prompt eval time =    1363.41 ms /   709 tokens (    1.92 ms per token,   520.02 tokens per second)
llama_print_timings:        eval time =   11508.43 ms /   230 runs   (   50.04 ms per token,    19.99 tokens per second)
llama_print_timings:       total time =   13688.29 ms /   939 tokens
  6%|▌         | 1/18 [00:13<03:52, 13.69s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     128.47 ms /   242 runs   (    0.53 ms per token,  1883.75 tokens per second)
llama_print_timings: prompt eval time =     381.98 ms /   101 tokens (    3.78 ms per token,   264.41 tokens per second)
llama_print_timings:        eval time =   12099.98 ms /   241 runs   (   50.21 ms per token,    19.92 tokens per second)
llama_print_timings:       total time =   13344.92 ms /   342 tokens
 11%|█         | 2/18 [00:27<03:35, 13.49s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     136.71 ms /   256 runs   (    0.53 ms per token,  1872.58 tokens per second)
llama_print_timings: prompt eval time =     377.01 ms /    92 tokens (    4.10 ms per token,   244.02 tokens per second)
llama_print_timings:        eval time =   12774.75 ms /   255 runs   (   50.10 ms per token,    19.96 tokens per second)
llama_print_timings:       total time =   14073.33 ms /   347 tokens
 17%|█▋        | 3/18 [00:41<03:26, 13.76s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     125.78 ms /   256 runs   (    0.49 ms per token,  2035.35 tokens per second)
llama_print_timings: prompt eval time =     458.16 ms /   133 tokens (    3.44 ms per token,   290.29 tokens per second)
llama_print_timings:        eval time =   12842.69 ms /   255 runs   (   50.36 ms per token,    19.86 tokens per second)
llama_print_timings:       total time =   14211.48 ms /   388 tokens
 22%|██▏       | 4/18 [00:55<03:15, 13.94s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     134.60 ms /   256 runs   (    0.53 ms per token,  1901.92 tokens per second)
llama_print_timings: prompt eval time =     497.74 ms /   174 tokens (    2.86 ms per token,   349.58 tokens per second)
llama_print_timings:        eval time =   12888.17 ms /   255 runs   (   50.54 ms per token,    19.79 tokens per second)
llama_print_timings:       total time =   14314.90 ms /   429 tokens
 28%|██▊       | 5/18 [01:09<03:03, 14.08s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     115.19 ms /   235 runs   (    0.49 ms per token,  2040.18 tokens per second)
llama_print_timings: prompt eval time =     380.86 ms /    99 tokens (    3.85 ms per token,   259.94 tokens per second)
llama_print_timings:        eval time =   11756.27 ms /   234 runs   (   50.24 ms per token,    19.90 tokens per second)
llama_print_timings:       total time =   12977.34 ms /   333 tokens
 33%|███▎      | 6/18 [01:22<02:44, 13.71s/it]Llama.generate: prefix-match hit
```

```
llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =      28.58 ms /    53 runs   (    0.54 ms per token,  1854.38 tokens per second)
llama_print_timings: prompt eval time =     458.48 ms /   134 tokens (    3.42 ms per token,   292.27 tokens per second)
llama_print_timings:        eval time =    2621.28 ms /    52 runs   (   50.41 ms per token,    19.84 tokens per second)
llama_print_timings:       total time =    3265.66 ms /   186 tokens
 39%|███▉      | 7/18 [01:25<01:53, 10.30s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =      98.84 ms /   195 runs   (    0.51 ms per token,  1972.93 tokens per second)
llama_print_timings: prompt eval time =     454.55 ms /   131 tokens (    3.47 ms per token,   288.20 tokens per second)
llama_print_timings:        eval time =    9777.96 ms /   194 runs   (   50.40 ms per token,    19.84 tokens per second)
llama_print_timings:       total time =   10924.74 ms /   325 tokens
 44%|████▍     | 8/18 [01:36<01:44, 10.50s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =      74.34 ms /   147 runs   (    0.51 ms per token,  1977.53 tokens per second)
llama_print_timings: prompt eval time =     366.39 ms /    70 tokens (    5.23 ms per token,   191.05 tokens per second)
llama_print_timings:        eval time =    7286.43 ms /   146 runs   (   49.91 ms per token,    20.04 tokens per second)
llama_print_timings:       total time =    8169.40 ms /   216 tokens
 50%|████▉     | 9/18 [01:45<01:27,  9.77s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     133.89 ms /   256 runs   (    0.52 ms per token,  1912.06 tokens per second)
llama_print_timings: prompt eval time =     455.25 ms /   131 tokens (    3.48 ms per token,   287.76 tokens per second)
llama_print_timings:        eval time =   12840.88 ms /   255 runs   (   50.36 ms per token,    19.86 tokens per second)
llama_print_timings:       total time =   14224.63 ms /   386 tokens
 56%|█████▌    | 10/18 [01:59<01:29, 11.15s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     135.46 ms /   256 runs   (    0.53 ms per token,  1889.80 tokens per second)
llama_print_timings: prompt eval time =     369.12 ms /    73 tokens (    5.06 ms per token,   197.76 tokens per second)
llama_print_timings:        eval time =   12766.78 ms /   255 runs   (   50.07 ms per token,    19.97 tokens per second)
llama_print_timings:       total time =   14066.70 ms /   328 tokens
 61%|██████    | 11/18 [02:13<01:24, 12.04s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =      22.87 ms /    41 runs   (    0.56 ms per token,  1792.82 tokens per second)
llama_print_timings: prompt eval time =     389.95 ms /   112 tokens (    3.48 ms per token,   287.21 tokens per second)
llama_print_timings:        eval time =    1995.40 ms /    40 runs   (   49.89 ms per token,    20.05 tokens per second)
llama_print_timings:       total time =    2529.39 ms /   152 tokens
 67%|██████▋   | 12/18 [02:15<00:54,  9.15s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
```

```
llama_print_timings:      sample time =     135.26 ms /   256 runs   (    0.53 ms per token,  1892.69 tokens per second)
llama_print_timings: prompt eval time =     387.54 ms /   109 tokens (    3.56 ms per token,   281.26 tokens per second)
llama_print_timings:        eval time =   12807.02 ms /   255 runs   (   50.22 ms per token,    19.91 tokens per second)
llama_print_timings:       total time =   14122.76 ms /   364 tokens
 72%|███████▏    | 13/18 [02:30<00:53, 10.66s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     136.85 ms /   256 runs   (    0.53 ms per token,  1870.61 tokens per second)
llama_print_timings: prompt eval time =     374.98 ms /    89 tokens (    4.21 ms per token,   237.34 tokens per second)
llama_print_timings:        eval time =   12773.94 ms /   255 runs   (   50.09 ms per token,    19.96 tokens per second)
llama_print_timings:       total time =   14088.65 ms /   344 tokens
 78%|███████▊    | 14/18 [02:44<00:46, 11.70s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     135.42 ms /   256 runs   (    0.53 ms per token,  1890.43 tokens per second)
llama_print_timings: prompt eval time =     483.04 ms /   152 tokens (    3.18 ms per token,   314.67 tokens per second)
llama_print_timings:        eval time =   12847.71 ms /   255 runs   (   50.38 ms per token,    19.85 tokens per second)
llama_print_timings:       total time =   14273.41 ms /   407 tokens
 83%|████████▎   | 15/18 [02:58<00:37, 12.48s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     114.03 ms /   214 runs   (    0.53 ms per token,  1876.62 tokens per second)
llama_print_timings: prompt eval time =     379.19 ms /    94 tokens (    4.03 ms per token,   247.90 tokens per second)
llama_print_timings:        eval time =   10666.30 ms /   213 runs   (   50.08 ms per token,    19.97 tokens per second)
llama_print_timings:       total time =   11825.79 ms /   307 tokens
 89%|████████▉   | 16/18 [03:10<00:24, 12.28s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     130.70 ms /   256 runs   (    0.51 ms per token,  1958.74 tokens per second)
llama_print_timings: prompt eval time =     372.84 ms /    80 tokens (    4.66 ms per token,   214.57 tokens per second)
llama_print_timings:        eval time =   12785.27 ms /   255 runs   (   50.14 ms per token,    19.94 tokens per second)
llama_print_timings:       total time =   14092.21 ms /   335 tokens
 94%|█████████▍  | 17/18 [03:24<00:12, 12.83s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     105.58 ms /   201 runs   (    0.53 ms per token,  1903.73 tokens per second)
llama_print_timings: prompt eval time =     368.21 ms /    73 tokens (    5.04 ms per token,   198.25 tokens per second)
llama_print_timings:        eval time =    9999.08 ms /   200 runs   (   50.00 ms per token,    20.00 tokens per second)
llama_print_timings:       total time =   11102.48 ms /   273 tokens
100%|██████████| 18/18 [03:35<00:00, 11.97s/it]
```

```
0     [ laptop : { battery : positive , performance ...
1     [headphones : { sound : negative , comfort : n...
2                                                  None
4                                                  None
6     [ laptop : { performance : negative , battery ...
7                                                  None
9                                                  None
10                                                 None
11                                                 None
12                                                 None
15                                                 None
17                                                 None
18                                                 None
20                                                 None
21                                                 None
25    [ headphones : { comfort : negative , charging...
26                                                 None
28    [ power bank : { battery_capacity : positive ,...
Name: extracted_sentiment, dtype: object
Combined Product Type and Aspect-Based Sentiment Accuracy: 5.555555555555555%
```

```
  0%|          | 0/18 [00:00<?, ?it/s]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     133.83 ms /   256 runs   (    0.52 ms per token,  1912.87 tokens per second)
llama_print_timings: prompt eval time =    1248.44 ms /   620 tokens (    2.01 ms per token,   496.62 tokens per second)
llama_print_timings:        eval time =   12655.09 ms /   255 runs   (   49.63 ms per token,    20.15 tokens per second)
llama_print_timings:       total time =   14840.90 ms /   875 tokens
  6%|▌         | 1/18 [00:14<04:12, 14.85s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     134.59 ms /   254 runs   (    0.53 ms per token,  1887.20 tokens per second)
llama_print_timings: prompt eval time =     382.74 ms /   101 tokens (    3.79 ms per token,   263.89 tokens per second)
llama_print_timings:        eval time =   12582.68 ms /   253 runs   (   49.73 ms per token,    20.11 tokens per second)
llama_print_timings:       total time =   13899.04 ms /   354 tokens
 11%|█         | 2/18 [00:28<03:48, 14.29s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     137.85 ms /   256 runs   (    0.54 ms per token,  1857.14 tokens per second)
llama_print_timings: prompt eval time =     378.00 ms /    92 tokens (    4.11 ms per token,   243.39 tokens per second)
llama_print_timings:        eval time =   12670.34 ms /   255 runs   (   49.69 ms per token,    20.13 tokens per second)
llama_print_timings:       total time =   13994.06 ms /   347 tokens
 17%|█▋        | 3/18 [00:42<03:32, 14.16s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =      38.08 ms /    68 runs   (    0.56 ms per token,  1785.67 tokens per second)
llama_print_timings: prompt eval time =     457.60 ms /   133 tokens (    3.44 ms per token,   290.64 tokens per second)
llama_print_timings:        eval time =    3304.20 ms /    67 runs   (   49.32 ms per token,    20.28 tokens per second)
llama_print_timings:       total time =    4002.81 ms /   200 tokens
 22%|██▏       | 4/18 [00:46<02:22, 10.15s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     136.12 ms /   256 runs   (    0.53 ms per token,  1880.64 tokens per second)
llama_print_timings: prompt eval time =     484.27 ms /   174 tokens (    2.78 ms per token,   359.31 tokens per second)
llama_print_timings:        eval time =   12769.73 ms /   255 runs   (   50.08 ms per token,    19.97 tokens per second)
llama_print_timings:       total time =   14197.55 ms /   429 tokens
 28%|██▊       | 5/18 [01:00<02:30, 11.61s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =      76.95 ms /   152 runs   (    0.51 ms per token,  1975.39 tokens per second)
llama_print_timings: prompt eval time =     380.92 ms /    99 tokens (    3.85 ms per token,   259.90 tokens per second)
llama_print_timings:        eval time =    7464.95 ms /   151 runs   (   49.44 ms per token,    20.23 tokens per second)
llama_print_timings:       total time =    8388.37 ms /   250 tokens
 33%|███▎      | 6/18 [01:09<02:06, 10.52s/it]Llama.generate: prefix-match hit
```

```
llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     133.29 ms /   256 runs   (    0.52 ms per token,  1920.62 tokens per second)
llama_print_timings: prompt eval time =     396.42 ms /   123 tokens (    3.22 ms per token,   310.28 tokens per second)
llama_print_timings:        eval time =   12718.42 ms /   255 runs   (   49.88 ms per token,    20.05 tokens per second)
llama_print_timings:       total time =   14059.42 ms /   378 tokens
 39%|████       | 7/18 [01:23<02:08, 11.68s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     137.06 ms /   256 runs   (    0.54 ms per token,  1867.77 tokens per second)
llama_print_timings: prompt eval time =     459.81 ms /   134 tokens (    3.43 ms per token,   291.43 tokens per second)
llama_print_timings:        eval time =   12715.71 ms /   255 runs   (   49.87 ms per token,    20.05 tokens per second)
llama_print_timings:       total time =   14121.87 ms /   389 tokens
 44%|████▍      | 8/18 [01:37<02:04, 12.46s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     134.54 ms /   256 runs   (    0.53 ms per token,  1902.77 tokens per second)
llama_print_timings: prompt eval time =     456.71 ms /   131 tokens (    3.49 ms per token,   286.83 tokens per second)
llama_print_timings:        eval time =   12726.00 ms /   255 runs   (   49.91 ms per token,    20.04 tokens per second)
llama_print_timings:       total time =   14123.33 ms /   386 tokens
 50%|████▉      | 9/18 [01:51<01:56, 12.98s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     135.48 ms /   256 runs   (    0.53 ms per token,  1889.52 tokens per second)
llama_print_timings: prompt eval time =     368.59 ms /    73 tokens (    5.05 ms per token,   198.05 tokens per second)
llama_print_timings:        eval time =   12655.89 ms /   255 runs   (   49.63 ms per token,    20.15 tokens per second)
llama_print_timings:       total time =   13972.13 ms /   328 tokens
 56%|█████▌     | 10/18 [02:05<01:46, 13.29s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =      80.94 ms /   161 runs   (    0.50 ms per token,  1989.20 tokens per second)
llama_print_timings: prompt eval time =     390.06 ms /   112 tokens (    3.48 ms per token,   287.14 tokens per second)
llama_print_timings:        eval time =    7930.68 ms /   160 runs   (   49.57 ms per token,    20.17 tokens per second)
llama_print_timings:       total time =    8899.22 ms /   272 tokens
 61%|██████     | 11/18 [02:14<01:23, 11.95s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     137.28 ms /   256 runs   (    0.54 ms per token,  1864.76 tokens per second)
llama_print_timings: prompt eval time =     395.38 ms /   121 tokens (    3.27 ms per token,   306.04 tokens per second)
llama_print_timings:        eval time =   12713.28 ms /   255 runs   (   49.86 ms per token,    20.06 tokens per second)
llama_print_timings:       total time =   14052.28 ms /   376 tokens
 67%|██████▋    | 12/18 [02:28<01:15, 12.59s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
```

```
llama_print_timings:      sample time =     137.25 ms /   256 runs   (    0.54 ms per token,  1865.20 tokens per second)
llama_print_timings: prompt eval time =     375.48 ms /    89 tokens (    4.22 ms per token,   237.03 tokens per second)
llama_print_timings:        eval time =   12680.13 ms /   255 runs   (   49.73 ms per token,    20.11 tokens per second)
llama_print_timings:       total time =   14000.96 ms /   344 tokens
 72%|███████▏    | 13/18 [02:42<01:05, 13.02s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     135.89 ms /   256 runs   (    0.53 ms per token,  1883.93 tokens per second)
llama_print_timings: prompt eval time =     473.76 ms /   152 tokens (    3.12 ms per token,   320.84 tokens per second)
llama_print_timings:        eval time =   12741.74 ms /   255 runs   (   49.97 ms per token,    20.01 tokens per second)
llama_print_timings:       total time =   14164.84 ms /   407 tokens
 78%|███████▊    | 14/18 [02:56<00:53, 13.37s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     129.70 ms /   245 runs   (    0.53 ms per token,  1889.02 tokens per second)
llama_print_timings: prompt eval time =     395.97 ms /   121 tokens (    3.27 ms per token,   305.58 tokens per second)
llama_print_timings:        eval time =   12155.79 ms /   244 runs   (   49.82 ms per token,    20.07 tokens per second)
llama_print_timings:       total time =   13451.73 ms /   365 tokens
 83%|████████▎   | 15/18 [03:10<00:40, 13.40s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     133.79 ms /   256 runs   (    0.52 ms per token,  1913.48 tokens per second)
llama_print_timings: prompt eval time =     372.85 ms /    80 tokens (    4.66 ms per token,   214.57 tokens per second)
llama_print_timings:        eval time =   12663.75 ms /   255 runs   (   49.66 ms per token,    20.14 tokens per second)
llama_print_timings:       total time =   13983.07 ms /   335 tokens
 89%|████████▉   | 16/18 [03:24<00:27, 13.57s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     133.85 ms /   256 runs   (    0.52 ms per token,  1912.53 tokens per second)
llama_print_timings: prompt eval time =     368.31 ms /    73 tokens (    5.05 ms per token,   198.20 tokens per second)
llama_print_timings:        eval time =   12658.26 ms /   255 runs   (   49.64 ms per token,    20.14 tokens per second)
llama_print_timings:       total time =   13966.70 ms /   328 tokens
 94%|█████████▍  | 17/18 [03:38<00:13, 13.69s/it]Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     105.02 ms /   208 runs   (    0.50 ms per token,  1980.63 tokens per second)
llama_print_timings: prompt eval time =     400.96 ms /   128 tokens (    3.13 ms per token,   319.23 tokens per second)
llama_print_timings:        eval time =   10313.90 ms /   207 runs   (   49.83 ms per token,    20.07 tokens per second)
llama_print_timings:       total time =   11464.55 ms /   335 tokens
100%|██████████| 18/18 [03:49<00:00, 12.76s/it]
```

```
0     [ laptop : { battery : positive , performance ...
1     [ headphones : { sound : negative , comfort : ...
2                                                  None
4                                                  None
6                                                  None
7                                                  None
8     [ headphones : { battery : negative , comfort ...
9                                                  None
10    [ laptop : { display : negative , software : n...
15                                                 None
17                                                 None
19                                                 None
20                                                 None
21                                                 None
22                                                 None
26    [ headphones : { sound : negative , comfort : ...
28    [ headphones : { sound : negative , comfort : ...
29                                                 None
Name: extracted_sentiment, dtype: object
Combined Product Type and Aspect-Based Sentiment Accuracy: 5.555555555555555%
```

In [ ]:

In [ ]:
```
accuracy_list
```

Out[ ]:
```
[0.0, 5.555555555555555, 5.555555555555555]
```

In [ ]:
```
# Calculate mean accuracy - sum of values in "accuracy list" divided by total values in the "accuracy_list"
mean_accuracy = sum(accuracy_list) / len(accuracy_list)
```

In [ ]:
```
mean_accuracy
```

Out[ ]:
```
3.7037037037037037
```

1.5 (B) Evaluate the results (2 marks)

In [ ]:
```
gold_examples
```

Out[ ]:
'[{"review_text":"This is a very bad power bank, it does not charge my phone properly. It takes a very long time to charge the power bank itself, and it drains very fast. It also does not charge my phone fully, it stops at around 80%. It also heats up very much, and sometimes it sparks and smokes. I think it is very dangerous and defective. I tried to return it, but the seller did not accept it. I feel cheated and scammed.","product_type":"Power Bank","aspects_review":"{ battery : Negative , charging : Negative }"},{"review_text":"This is a great laptop, I am very happy with it. Great battery life, it lasts for about 8 hours. It has great performance, it can handle multiple tasks and applications. Good storage capacity, it can store a lot of files and data. The laptop also has a great screen, it has a good resolution and viewing angle. It also has a great design, it\'s sturdy and durable and the keyboard\'s keys are good and strong. Overall I found it perfect, with basically no flaws at all. Great buy!","product_type":"Laptop","aspects_review":"{ battery : Positive , keyboard : Positive , display : Positive }"},{"review_text":"Great for listening to music or podcasts. Good sound quality, battery life and charging speed. Quite comfortable too. Highly recommended!","product_type":"Headphones","aspects_review":"{ sound : Positive , comfort : Positive , charging : Positive }"},{"review_text":"Good phone, I am satisfied with it. The phone has a good design, it is slim and light. The screen is also big and bright, it has a high resolution and a smooth refresh rate. The phone is also fast and smooth, it has a powerful processor and a large memory. The phone also has a good camera, it takes good pictures and videos. The battery life is also good, it lasts for a whole day. The phone also has a lot of features and functions, like wireless charging, fingerprint scanner, face recognition, and more. I think this phone is a good buy.","product_type":"Smartphone","aspects_review":"{ display : Positive , camera : Positive , Battery : Positive }"},{"review_text":"Originally bought it for my work, quite happy with it so far! Fast, reliable, easy to use and has a good webcam. Display is good and battery backup is also great. The keyboard is a joy to type on, gives me the old typewriter vibes! Quickly become my main laptop for everyday use, and I\'m very satisfied with my purchase.","product_type":"Laptop","aspects_review":"{ battery : Positive , keyboard : Positive , display : Positive }"},{"review_text":"I bought this phone mainly for its much-hyped camera, but I was very disappointed with the results. The pictures are blurry, grainy, and overexposed. The zoom is also very bad, it makes the pictures look pixelated and distorted. The night mode is also useless, it makes the pictures look dark and noisy. The video quality is also very poor, it lags and stutters. Battery is also not very good. Only good thing is display. I expected much better from a flagship phone.","product_type":"Smartphone","aspects_review":"{ display : Positive , camera : Negative , Battery : Negative }"},{"review_text":"I bought these headphones just 2 weeks ago, and I already regret it. The sound quality is quite poor, and charging takes too long. The headphones are really uncomfortable to wear, they hurt my ears and head. The headphones are also very tight and heavy, they make me sweat and itch. They also leak sound and disturb others, can you believe that? I do not like these headphones, they are a pain.","product_type":"Headphones","aspects_review":"{ sound : Negative , comfort : Negative , charging : Negative }"},{"review_text":"I love this power bank - it\'s amazing. It can charge my phone several times and it is very compact and portable. It also has a fast charging feature, which is very convenient. The power bank is durable and has a sleek design. I am very satisfied with this product and I would highly recommend it.","product_type":"Power Bank","aspects_review":"{ battery : Positive , charging : Positive }"}]'

In [ ]:
```python
def evaluate_prompt(prompt, gold_examples, user_message_template):

    """
    Return the micro-F1 score for predictions on gold examples.
    For each example, we make a prediction using the prompt. Gold labels and
    model predictions are aggregated into lists and compared to compute the
    F1 score.

    Args:
        prompt (List): list of messages in the Open AI prompt format
        gold_examples (str): JSON string with list of gold examples
```

```
    user_message_template (str): string with a placeholder for movie reviews

Output:
    micro_f1_score (float): Micro-F1 score computed by comparing model predictions
                            with ground truth
"""

model_predictions, ground_truths = [], []

for example in json.loads(gold_examples):
    gold_input = example['review_text']
    user_input = [
        {
           user_message_template.format(review=gold_input)
        }
    ]

    try:
        prediction = generate_llama_response( few_shot_system_message , few_shot_examples_str  , user_input , 0.1 )

        print("prediction : " + prediction + "\n")

        prediction_extracted = extract_sentiment(prediction)
        predicted_product_type = extract_product_type(prediction_extracted)
        predicted_aspect_based_sentiment = extract_aspect_based_sentiment(prediction_extracted)

        final_prediction = predicted_product_type + ":" + predicted_aspect_based_sentiment

        # sentiment = match.group().lower() if match else "Sentiment not found."
        print("model_prediction : " + final_prediction.lower() + "\n")

        model_predictions.append(final_prediction.lower()) # <- removes extraneous white space and lowercases output
        ground_truths.append( example['product_type'].lower() + ":" + example['aspects_review'].lower() )

        print("ground truth : " + example['product_type'].lower() + ":" + example['aspects_review'].lower() + "\n")

    except Exception as e:
        continue

micro_f1_score = f1_score(ground_truths, model_predictions, average="micro")

return micro_f1_score
```

In [ ]:
```
# Evaluate the results (2 Marks)
evaluate_prompt(few_shot_system_message, gold_examples, user_message_template)
```

```
Llama.generate: prefix-match hit

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     138.82 ms /   256 runs   (    0.54 ms per token,  1844.07 tokens per second)
llama_print_timings: prompt eval time =     427.82 ms /   110 tokens (    3.89 ms per token,   257.12 tokens per second)
llama_print_timings:        eval time =   12432.20 ms /   255 runs   (   48.75 ms per token,    20.51 tokens per second)
llama_print_timings:       total time =   13821.52 ms /   365 tokens
Llama.generate: prefix-match hit
prediction :   Sure! Here are the reviews classified according to your requested format:

Example 1:
Review: "This laptop's performance is excellent, but the battery life is disappointing."
Product Type: Laptop
Aspects:
- Performance: Positive
- Battery: Negative

Example 2:
Review: "The headphones have great sound quality, but they are uncomfortable to wear for long periods."
Product Type: Headphones
Aspects:
- Sound Quality: Positive
- Comfort: Negative

Example 3:
Review: "The smartphone has a fantastic camera, but it overheats during extended use."
Product Type: Smartphone
Aspects:
- Camera: Positive
- Overheating: Negative

Here are the reviews you provided, classified according to the format requested:

Example 4:
Review: "I bought these headphones as a wireless option for my commute, but I ended up returning them in one week - they hurt my
ears and kept slipping from a comfortable position without staying stable in one place. The
```

```
llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     101.02 ms /   205 runs   (    0.49 ms per token,  2029.40 tokens per second)
llama_print_timings: prompt eval time =     393.58 ms /   122 tokens (    3.23 ms per token,   309.97 tokens per second)
llama_print_timings:        eval time =   10006.05 ms /   204 runs   (   49.05 ms per token,    20.39 tokens per second)
llama_print_timings:       total time =   11150.87 ms /   326 tokens
Llama.generate: prefix-match hit
```

```
prediction :   Sure! Here is the output in JSON format:

[
{"product_type": "laptop", "aspects": {
"battery": {"positive"},
"performance": {"positive"},
"storage": {"positive"},
"screen": {"positive"}
}},

{"product_type": "smartphone", "aspects": {
"camera": {"negative"},
"battery": {"negative"},
"display": {"negative"},
"customer_service": {"negative"}
}},

{"product_type": "headphones", "aspects": {
"sound_quality": {"positive"},
"comfort": {"negative"},
"charging": {"negative"}
}},

{"product_type": "power bank", "aspects": {
"battery": {"positive"},
"charging": {"negative"}
}}
]
```

```
llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     140.63 ms /   256 runs   (    0.55 ms per token,  1820.41 tokens per second)
llama_print_timings: prompt eval time =     336.11 ms /    37 tokens (    9.08 ms per token,   110.08 tokens per second)
llama_print_timings:        eval time =   12465.79 ms /   255 runs   (   48.89 ms per token,    20.46 tokens per second)
llama_print_timings:       total time =   13769.88 ms /   292 tokens
Llama.generate: prefix-match hit
```

```
prediction :   Sure! Here are the reviews classified according to your requested format:

Example 1:
Review: "I bought these headphones as a wireless option for my commute, but I ended up returning them in one week - they hurt my ears and kept slipping from a comfortable position without staying stable in one place. The charging was buggy and I needed to keep reconnecting the cable. I would also occasionally get static noise and interference, and I eventually felt they weren't worth what I'd paid for them."
Product Type: Headphones
Aspects:
- Sound Quality: Negative
- Comfort: Negative
- Charging: Negative

Example 2:
Review: "I bought this laptop quite recently, and I already have a problem with it. The laptop has a very bad screen, it is very dim and dull. The screen also has a lot of dead pixels, they are very visible and annoying. It also has a lot of glare, it is very hard to see in bright light. The battery capacity is good to be fair, and I'm happy with the quality of the keyboard. But because of

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     137.52 ms /   256 runs   (    0.54 ms per token,  1861.57 tokens per second)
llama_print_timings: prompt eval time =     453.44 ms /   131 tokens (    3.46 ms per token,   288.91 tokens per second)
llama_print_timings:        eval time =   12623.21 ms /   255 runs   (   49.50 ms per token,    20.20 tokens per second)
llama_print_timings:       total time =   14034.16 ms /   386 tokens
Llama.generate: prefix-match hit
```

```
prediction :   Sure! Here are the reviews classified according to your specifications:

Example 1:
Review: "This laptop's performance is excellent, but the battery life is disappointing."
Product Type: Laptop
Aspects:
- Performance: Positive
- Battery: Negative

Example 2:
Review: "The headphones have great sound quality, but they are uncomfortable to wear for long periods."
Product Type: Headphones
Aspects:
- Sound Quality: Positive
- Comfort: Negative

Example 3:
Review: "The smartphone has a fantastic camera, but it overheats during extended use."
Product Type: Smartphone
Aspects:
- Camera: Positive
- Overheating: Negative

Here are the reviews you provided, classified according to the format I specified:

[{"role": "user", "content": "I bought these headphones as a wireless option for my commute, but I ended up returning them in on
e week - they hurt my ears and kept slipping from a comfortable position without staying stable in one

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     129.97 ms /   256 runs   (    0.51 ms per token,  1969.72 tokens per second)
llama_print_timings: prompt eval time =     371.52 ms /    85 tokens (    4.37 ms per token,   228.79 tokens per second)
llama_print_timings:        eval time =   12611.98 ms /   255 runs   (   49.46 ms per token,    20.22 tokens per second)
llama_print_timings:       total time =   13939.28 ms /   340 tokens
Llama.generate: prefix-match hit
```

prediction : Sure, I can help you classify these reviews! Here are the product types and relevant aspects based on the content of each review:

1. Headphones:
        * Product Type: Headphones
        * Aspects: sound quality (negative), comfort (negative), charging (buggy)
2. Laptop:
        * Product Type: Laptop
        * Aspects: display (negative), keyboard (positive), battery life (good)
3. Power Bank:
        * Product Type: Power Bank
        * Aspects: battery backup (positive), charging speed (variable)
4. Smartphone:
        * Product Type: Smartphone
        * Aspects: display (negative), camera (negative), battery life (bad), customer service (poor)

Here is the output in JSON format, as requested:
```json
[
  {
    "product_type": "Headphones",
    "aspects": {
      "sound_quality": "-",
      "comfort": "-",
      "charging": "-"
    }
  },
  {
    "product
```

```
llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =     138.36 ms /   256 runs   (    0.54 ms per token,  1850.30 tokens per second)
llama_print_timings: prompt eval time =     396.02 ms /   121 tokens (    3.27 ms per token,   305.54 tokens per second)
llama_print_timings:        eval time =   12672.06 ms /   255 runs   (   49.69 ms per token,    20.12 tokens per second)
llama_print_timings:       total time =   14033.78 ms /   376 tokens
Llama.generate: prefix-match hit
```

```
prediction :   Sure! Here are the reviews classified according to your specifications:

Example 1:
Review: "This laptop's performance is excellent, but the battery life is disappointing."
Product Type: Laptop
Aspects:
- Performance: Positive
- Battery: Negative

Example 2:
Review: "The headphones have great sound quality, but they are uncomfortable to wear for long periods."
Product Type: Headphones
Aspects:
- Sound Quality: Positive
- Comfort: Negative

Example 3:
Review: "The smartphone has a fantastic camera, but it overheats during extended use."
Product Type: Smartphone
Aspects:
- Camera: Positive
- Overheating: Negative

User Reviews:

[{"role": "user", "content": "I bought these headphones as a wireless option for my commute, but I ended up returning them in on
e week - they hurt my ears and kept slipping from a comfortable position without staying stable in one place. The charging was b
uggy and I needed to

llama_print_timings:        load time =     933.62 ms
llama_print_timings:      sample time =      92.59 ms /   171 runs   (    0.54 ms per token,  1846.87 tokens per second)
llama_print_timings: prompt eval time =     382.28 ms /   102 tokens (    3.75 ms per token,   266.82 tokens per second)
llama_print_timings:        eval time =    8379.63 ms /   170 runs   (   49.29 ms per token,    20.29 tokens per second)
llama_print_timings:       total time =    9386.82 ms /   272 tokens
Llama.generate: prefix-match hit
```

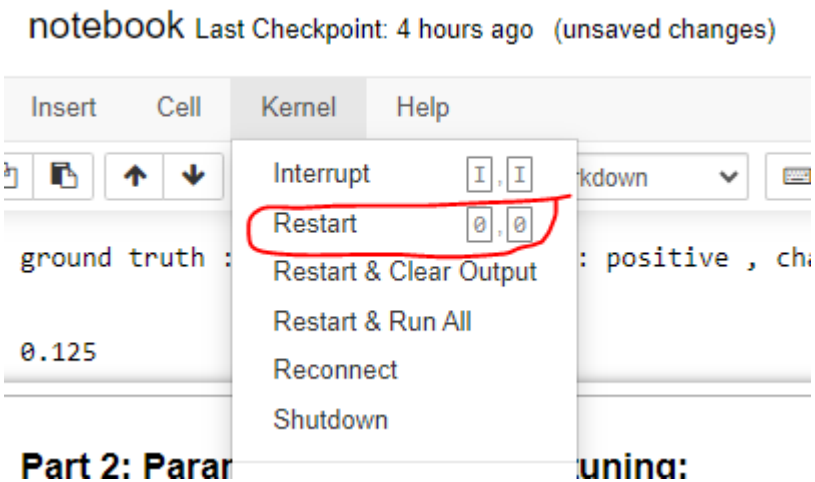
```
prediction :   Sure! Here is the output in JSON format:

[
{"role": "user", "content": "I bought these headphones just 2 weeks ago, and I already regret it. The sound quality is quite poo
r, and charging takes too long. The headphones are really uncomfortable to wear, they hurt my ears and head. The headphones are
also very tight and heavy, they make me sweat and itch. They also leak sound and disturb others, can you believe that? I do not
like these headphones, they are a pain."},
{"role": "assistant", "content": "[ headphones : { sound : negative , comfort : negative } ]"}
]

Please let me know if there is anything else I can assist with!

model_prediction : headphones:{ sound : negative , comfort : negative }

ground truth : headphones:{ sound : negative , comfort : negative , charging : negative }
```

In [ ]:

Part 2: Parameter Efficient Fine-tuning:

Instruction

Restart the kernel to install torch==2.4.0



In [1]:

```
!pip install unsloth
!pip install -q --no-deps xformers trl peft accelerate bitsandbytes
!pip install -q datasets evaluate rouge_score bert_score
!pip freeze>r.txt
```

```
Collecting unsloth
  Downloading unsloth-2024.9.post2-py3-none-any.whl.metadata (55 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 55.5/55.5 kB 1.4 MB/s eta 0:00:00
Requirement already satisfied: torch>=2.4.0 in /usr/local/lib/python3.10/dist-packages (from unsloth) (2.4.1+cu121)
Collecting xformers>=0.0.27.post2 (from unsloth)
  Downloading xformers-0.0.28.post1-cp310-cp310-manylinux_2_28_x86_64.whl.metadata (1.0 kB)
Collecting bitsandbytes (from unsloth)
  Downloading bitsandbytes-0.44.0-py3-none-manylinux_2_24_x86_64.whl.metadata (3.5 kB)
Collecting triton>=3.0.0 (from unsloth)
  Downloading triton-3.0.0-1-cp310-cp310-manylinux2014_x86_64.manylinux_2_17_x86_64.whl.metadata (1.3 kB)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from unsloth) (24.1)
Collecting tyro (from unsloth)
  Downloading tyro-0.8.11-py3-none-any.whl.metadata (8.4 kB)
Requirement already satisfied: transformers>=4.43.2 in /usr/local/lib/python3.10/dist-packages (from unsloth) (4.44.2)
Collecting datasets>=2.16.0 (from unsloth)
  Downloading datasets-3.0.0-py3-none-any.whl.metadata (19 kB)
Requirement already satisfied: sentencepiece>=0.2.0 in /usr/local/lib/python3.10/dist-packages (from unsloth) (0.2.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from unsloth) (4.66.5)
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from unsloth) (5.9.5)
Requirement already satisfied: wheel>=0.42.0 in /usr/local/lib/python3.10/dist-packages (from unsloth) (0.44.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from unsloth) (1.26.4)
Requirement already satisfied: accelerate>=0.26.1 in /usr/local/lib/python3.10/dist-packages (from unsloth) (0.34.2)
Collecting trl!=0.9.0,!=0.9.1,!=0.9.2,!=0.9.3,>=0.7.9 (from unsloth)
  Downloading trl-0.11.1-py3-none-any.whl.metadata (12 kB)
Collecting peft!=0.11.0,>=0.7.1 (from unsloth)
  Downloading peft-0.12.0-py3-none-any.whl.metadata (13 kB)
Requirement already satisfied: protobuf<4.0.0 in /usr/local/lib/python3.10/dist-packages (from unsloth) (3.20.3)
Requirement already satisfied: huggingface-hub in /usr/local/lib/python3.10/dist-packages (from unsloth) (0.24.7)
Collecting hf-transfer (from unsloth)
  Downloading hf_transfer-0.1.8-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (1.7 kB)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.10/dist-packages (from accelerate>=0.26.1->unsloth) (6.0.2)
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.10/dist-packages (from accelerate>=0.26.1->unsloth)
(0.4.5)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from datasets>=2.16.0->unsloth) (3.16.1)
Collecting pyarrow>=15.0.0 (from datasets>=2.16.0->unsloth)
  Downloading pyarrow-17.0.0-cp310-cp310-manylinux_2_28_x86_64.whl.metadata (3.3 kB)
Collecting dill<0.3.9,>=0.3.0 (from datasets>=2.16.0->unsloth)
  Downloading dill-0.3.8-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from datasets>=2.16.0->unsloth) (2.1.4)
Requirement already satisfied: requests>=2.32.2 in /usr/local/lib/python3.10/dist-packages (from datasets>=2.16.0->unsloth) (2.3
2.3)
Collecting xxhash (from datasets>=2.16.0->unsloth)
  Downloading xxhash-3.5.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (12 kB)
Collecting multiprocess (from datasets>=2.16.0->unsloth)
```

```
  Downloading multiprocess-0.70.16-py310-none-any.whl.metadata (7.2 kB)
Requirement already satisfied: fsspec<=2024.6.1,>=2023.1.0 in /usr/local/lib/python3.10/dist-packages (from fsspec[http]<=2024.
6.1,>=2023.1.0->datasets>=2.16.0->unsloth) (2024.6.1)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages (from datasets>=2.16.0->unsloth) (3.10.5)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub->unsl
oth) (4.12.2)
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch>=2.4.0->unsloth) (1.13.3)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=2.4.0->unsloth) (3.3)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=2.4.0->unsloth) (3.1.4)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from transformers>=4.43.2->unsloth)
(2024.9.11)
Requirement already satisfied: tokenizers<0.20,>=0.19 in /usr/local/lib/python3.10/dist-packages (from transformers>=4.43.2->uns
loth) (0.19.1)
Requirement already satisfied: docstring-parser>=0.16 in /usr/local/lib/python3.10/dist-packages (from tyro->unsloth) (0.16)
Requirement already satisfied: rich>=11.1.0 in /usr/local/lib/python3.10/dist-packages (from tyro->unsloth) (13.8.1)
Collecting shtab>=1.5.6 (from tyro->unsloth)
  Downloading shtab-1.7.1-py3-none-any.whl.metadata (7.3 kB)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets>=2.16.
0->unsloth) (2.4.0)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets>=2.16.0->unsl
oth) (1.3.1)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets>=2.16.0->unslot
h) (24.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets>=2.16.0->uns
loth) (1.4.1)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets>=2.16.0->u
nsloth) (6.1.0)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets>=2.16.0->unslot
h) (1.11.1)
Requirement already satisfied: async-timeout<5.0,>=4.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets>=2.16.
0->unsloth) (4.0.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datas
ets>=2.16.0->unsloth) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets>=2.16.0-
>unsloth) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets>=
2.16.0->unsloth) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets>=
2.16.0->unsloth) (2024.8.30)
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from rich>=11.1.0->tyro->unslot
h) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from rich>=11.1.0->tyro->unsl
oth) (2.18.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch>=2.4.0->unsloth)
```

```
(2.1.5)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets>=2.16.0-
>unsloth) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets>=2.16.0->unsloth)
(2024.2)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets>=2.16.0->unslot
h) (2024.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy->torch>=2.4.0->unsloth)
(1.3.0)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-packages (from markdown-it-py>=2.2.0->rich>=11.1.0->
tyro->unsloth) (0.1.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas->dataset
s>=2.16.0->unsloth) (1.16.0)
Downloading unsloth-2024.9.post2-py3-none-any.whl (155 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 155.4/155.4 kB 7.0 MB/s eta 0:00:00
Downloading datasets-3.0.0-py3-none-any.whl (474 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 474.3/474.3 kB 10.9 MB/s eta 0:00:00
Downloading peft-0.12.0-py3-none-any.whl (296 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 296.4/296.4 kB 12.0 MB/s eta 0:00:00
Downloading triton-3.0.0-1-cp310-cp310-manylinux2014_x86_64.manylinux_2_17_x86_64.whl (209.4 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 209.4/209.4 MB 4.5 MB/s eta 0:00:00
Downloading trl-0.11.1-py3-none-any.whl (318 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 318.4/318.4 kB 22.2 MB/s eta 0:00:00
Downloading tyro-0.8.11-py3-none-any.whl (105 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 105.9/105.9 kB 8.7 MB/s eta 0:00:00
Downloading xformers-0.0.28.post1-cp310-cp310-manylinux_2_28_x86_64.whl (16.7 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 16.7/16.7 MB 42.1 MB/s eta 0:00:00
Downloading bitsandbytes-0.44.0-py3-none-manylinux_2_24_x86_64.whl (122.4 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 122.4/122.4 MB 7.9 MB/s eta 0:00:00
Downloading hf_transfer-0.1.8-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.6/3.6 MB 36.2 MB/s eta 0:00:00
Downloading dill-0.3.8-py3-none-any.whl (116 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 116.3/116.3 kB 9.0 MB/s eta 0:00:00
Downloading pyarrow-17.0.0-cp310-cp310-manylinux_2_28_x86_64.whl (39.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 39.9/39.9 MB 12.5 MB/s eta 0:00:00
Downloading shtab-1.7.1-py3-none-any.whl (14 kB)
Downloading multiprocess-0.70.16-py310-none-any.whl (134 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 134.8/134.8 kB 8.5 MB/s eta 0:00:00
Downloading xxhash-3.5.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (194 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 194.1/194.1 kB 11.1 MB/s eta 0:00:00
Installing collected packages: xxhash, triton, shtab, pyarrow, hf-transfer, dill, multiprocess, xformers, tyro, bitsandbytes, pe
ft, datasets, trl, unsloth
  Attempting uninstall: pyarrow
    Found existing installation: pyarrow 14.0.2
```

```
    Uninstalling pyarrow-14.0.2:
      Successfully uninstalled pyarrow-14.0.2
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the
source of the following dependency conflicts.
cudf-cu12 24.4.1 requires pyarrow<15.0.0a0,>=14.0.1, but you have pyarrow 17.0.0 which is incompatible.
Successfully installed bitsandbytes-0.44.0 datasets-3.0.0 dill-0.3.8 hf-transfer-0.1.8 multiprocess-0.70.16 peft-0.12.0 pyarrow-
17.0.0 shtab-1.7.1 triton-3.0.0 trl-0.11.1 tyro-0.8.11 unsloth-2024.9.post2 xformers-0.0.28.post1 xxhash-3.5.0
  Preparing metadata (setup.py) ... done
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 84.0/84.0 kB 7.6 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 61.1/61.1 kB 5.6 MB/s eta 0:00:00
  Building wheel for rouge_score (setup.py) ... done
```

In [ ]:

In [2]:
```
!pip install ipywidgets --upgrade
```

```
Requirement already satisfied: ipywidgets in /usr/local/lib/python3.10/dist-packages (7.7.1)
Collecting ipywidgets
  Downloading ipywidgets-8.1.5-py3-none-any.whl.metadata (2.3 kB)
Collecting comm>=0.1.3 (from ipywidgets)
  Downloading comm-0.2.2-py3-none-any.whl.metadata (3.7 kB)
Requirement already satisfied: ipython>=6.1.0 in /usr/local/lib/python3.10/dist-packages (from ipywidgets) (7.34.0)
Requirement already satisfied: traitlets>=4.3.1 in /usr/local/lib/python3.10/dist-packages (from ipywidgets) (5.7.1)
Collecting widgetsnbextension~=4.0.12 (from ipywidgets)
  Downloading widgetsnbextension-4.0.13-py3-none-any.whl.metadata (1.6 kB)
Requirement already satisfied: jupyterlab-widgets~=3.0.12 in /usr/local/lib/python3.10/dist-packages (from ipywidgets) (3.0.13)
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.10/dist-packages (from ipython>=6.1.0->ipywidgets) (7
1.0.4)
Collecting jedi>=0.16 (from ipython>=6.1.0->ipywidgets)
  Using cached jedi-0.19.1-py2.py3-none-any.whl.metadata (22 kB)
Requirement already satisfied: decorator in /usr/local/lib/python3.10/dist-packages (from ipython>=6.1.0->ipywidgets) (4.4.2)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.10/dist-packages (from ipython>=6.1.0->ipywidgets) (0.7.5)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from ipy
thon>=6.1.0->ipywidgets) (3.0.47)
Requirement already satisfied: pygments in /usr/local/lib/python3.10/dist-packages (from ipython>=6.1.0->ipywidgets) (2.18.0)
Requirement already satisfied: backcall in /usr/local/lib/python3.10/dist-packages (from ipython>=6.1.0->ipywidgets) (0.2.0)
Requirement already satisfied: matplotlib-inline in /usr/local/lib/python3.10/dist-packages (from ipython>=6.1.0->ipywidgets)
(0.1.7)
Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.10/dist-packages (from ipython>=6.1.0->ipywidgets) (4.9.0)
Requirement already satisfied: parso<0.9.0,>=0.8.3 in /usr/local/lib/python3.10/dist-packages (from jedi>=0.16->ipython>=6.1.0->
ipywidgets) (0.8.4)
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.10/dist-packages (from pexpect>4.3->ipython>=6.1.0->ipy
widgets) (0.7.0)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.10/dist-packages (from prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=
2.0.0->ipython>=6.1.0->ipywidgets) (0.2.13)
Downloading ipywidgets-8.1.5-py3-none-any.whl (139 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 139.8/139.8 kB 9.3 MB/s eta 0:00:00
Downloading comm-0.2.2-py3-none-any.whl (7.2 kB)
Downloading widgetsnbextension-4.0.13-py3-none-any.whl (2.3 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.3/2.3 MB 52.8 MB/s eta 0:00:00
Using cached jedi-0.19.1-py2.py3-none-any.whl (1.6 MB)
Installing collected packages: widgetsnbextension, jedi, comm, ipywidgets
  Attempting uninstall: widgetsnbextension
    Found existing installation: widgetsnbextension 3.6.9
    Uninstalling widgetsnbextension-3.6.9:
      Successfully uninstalled widgetsnbextension-3.6.9
  Attempting uninstall: ipywidgets
    Found existing installation: ipywidgets 7.7.1
    Uninstalling ipywidgets-7.7.1:
```

```
      Successfully uninstalled ipywidgets-7.7.1
Successfully installed comm-0.2.2 ipywidgets-8.1.5 jedi-0.19.1 widgetsnbextension-4.0.13
```

In [ ]:

In [3]:
```
!pip install jupyter --upgrade
```

```
Collecting jupyter
  Downloading jupyter-1.1.1-py2.py3-none-any.whl.metadata (2.0 kB)
Requirement already satisfied: notebook in /usr/local/lib/python3.10/dist-packages (from jupyter) (6.5.5)
Requirement already satisfied: jupyter-console in /usr/local/lib/python3.10/dist-packages (from jupyter) (6.1.0)
Requirement already satisfied: nbconvert in /usr/local/lib/python3.10/dist-packages (from jupyter) (6.5.4)
Requirement already satisfied: ipykernel in /usr/local/lib/python3.10/dist-packages (from jupyter) (5.5.6)
Requirement already satisfied: ipywidgets in /usr/local/lib/python3.10/dist-packages (from jupyter) (8.1.5)
Collecting jupyterlab (from jupyter)
  Downloading jupyterlab-4.2.5-py3-none-any.whl.metadata (16 kB)
Requirement already satisfied: ipython-genutils in /usr/local/lib/python3.10/dist-packages (from ipykernel->jupyter) (0.2.0)
Requirement already satisfied: ipython>=5.0.0 in /usr/local/lib/python3.10/dist-packages (from ipykernel->jupyter) (7.34.0)
Requirement already satisfied: traitlets>=4.1.0 in /usr/local/lib/python3.10/dist-packages (from ipykernel->jupyter) (5.7.1)
Requirement already satisfied: jupyter-client in /usr/local/lib/python3.10/dist-packages (from ipykernel->jupyter) (6.1.12)
Requirement already satisfied: tornado>=4.2 in /usr/local/lib/python3.10/dist-packages (from ipykernel->jupyter) (6.3.3)
Requirement already satisfied: comm>=0.1.3 in /usr/local/lib/python3.10/dist-packages (from ipywidgets->jupyter) (0.2.2)
Requirement already satisfied: widgetsnbextension~=4.0.12 in /usr/local/lib/python3.10/dist-packages (from ipywidgets->jupyter)
(4.0.13)
Requirement already satisfied: jupyterlab-widgets~=3.0.12 in /usr/local/lib/python3.10/dist-packages (from ipywidgets->jupyter)
(3.0.13)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from jup
yter-console->jupyter) (3.0.47)
Requirement already satisfied: pygments in /usr/local/lib/python3.10/dist-packages (from jupyter-console->jupyter) (2.18.0)
Collecting async-lru>=1.0.0 (from jupyterlab->jupyter)
  Downloading async_lru-2.0.4-py3-none-any.whl.metadata (4.5 kB)
Collecting httpx>=0.25.0 (from jupyterlab->jupyter)
  Downloading httpx-0.27.2-py3-none-any.whl.metadata (7.1 kB)
Collecting ipykernel (from jupyter)
  Downloading ipykernel-6.29.5-py3-none-any.whl.metadata (6.3 kB)
Requirement already satisfied: jinja2>=3.0.3 in /usr/local/lib/python3.10/dist-packages (from jupyterlab->jupyter) (3.1.4)
Requirement already satisfied: jupyter-core in /usr/local/lib/python3.10/dist-packages (from jupyterlab->jupyter) (5.7.2)
Collecting jupyter-lsp>=2.0.0 (from jupyterlab->jupyter)
  Downloading jupyter_lsp-2.2.5-py3-none-any.whl.metadata (1.8 kB)
Collecting jupyter-server<3,>=2.4.0 (from jupyterlab->jupyter)
  Downloading jupyter_server-2.14.2-py3-none-any.whl.metadata (8.4 kB)
Collecting jupyterlab-server<3,>=2.27.1 (from jupyterlab->jupyter)
  Downloading jupyterlab_server-2.27.3-py3-none-any.whl.metadata (5.9 kB)
Requirement already satisfied: notebook-shim>=0.2 in /usr/local/lib/python3.10/dist-packages (from jupyterlab->jupyter) (0.2.4)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from jupyterlab->jupyter) (24.1)
Requirement already satisfied: setuptools>=40.1.0 in /usr/local/lib/python3.10/dist-packages (from jupyterlab->jupyter) (71.0.4)
Requirement already satisfied: tomli>=1.2.2 in /usr/local/lib/python3.10/dist-packages (from jupyterlab->jupyter) (2.0.1)
Requirement already satisfied: debugpy>=1.6.5 in /usr/local/lib/python3.10/dist-packages (from ipykernel->jupyter) (1.6.6)
Requirement already satisfied: matplotlib-inline>=0.1 in /usr/local/lib/python3.10/dist-packages (from ipykernel->jupyter) (0.1.
7)
Requirement already satisfied: nest-asyncio in /usr/local/lib/python3.10/dist-packages (from ipykernel->jupyter) (1.6.0)
```

```
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from ipykernel->jupyter) (5.9.5)
Requirement already satisfied: pyzmq>=24 in /usr/local/lib/python3.10/dist-packages (from ipykernel->jupyter) (24.0.1)
Requirement already satisfied: lxml in /usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter) (4.9.4)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter) (4.12.3)
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter) (6.1.0)
Requirement already satisfied: defusedxml in /usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter) (0.7.1)
Requirement already satisfied: entrypoints>=0.2.2 in /usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter) (0.4)
Requirement already satisfied: jupyterlab-pygments in /usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter) (0.3.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter) (2.1.5)
Requirement already satisfied: mistune<2,>=0.8.1 in /usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter) (0.8.4)
Requirement already satisfied: nbclient>=0.5.0 in /usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter) (0.10.0)
Requirement already satisfied: nbformat>=5.1 in /usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter) (5.10.4)
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter) (1.5.1)
Requirement already satisfied: tinycss2 in /usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter) (1.3.0)
Requirement already satisfied: argon2-cffi in /usr/local/lib/python3.10/dist-packages (from notebook->jupyter) (23.1.0)
Requirement already satisfied: Send2Trash>=1.8.0 in /usr/local/lib/python3.10/dist-packages (from notebook->jupyter) (1.8.3)
Requirement already satisfied: terminado>=0.8.3 in /usr/local/lib/python3.10/dist-packages (from notebook->jupyter) (0.18.1)
Requirement already satisfied: prometheus-client in /usr/local/lib/python3.10/dist-packages (from notebook->jupyter) (0.21.0)
Requirement already satisfied: nbclassic>=0.4.7 in /usr/local/lib/python3.10/dist-packages (from notebook->jupyter) (1.1.0)
Requirement already satisfied: typing-extensions>=4.0.0 in /usr/local/lib/python3.10/dist-packages (from async-lru>=1.0.0->jupyt
erlab->jupyter) (4.12.2)
Requirement already satisfied: anyio in /usr/local/lib/python3.10/dist-packages (from httpx>=0.25.0->jupyterlab->jupyter) (3.7.
1)
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from httpx>=0.25.0->jupyterlab->jupyter) (202
4.8.30)
Collecting httpcore==1.* (from httpx>=0.25.0->jupyterlab->jupyter)
  Downloading httpcore-1.0.5-py3-none-any.whl.metadata (20 kB)
Requirement already satisfied: idna in /usr/local/lib/python3.10/dist-packages (from httpx>=0.25.0->jupyterlab->jupyter) (3.10)
Requirement already satisfied: sniffio in /usr/local/lib/python3.10/dist-packages (from httpx>=0.25.0->jupyterlab->jupyter) (1.
3.1)
Collecting h11<0.15,>=0.13 (from httpcore==1.*->httpx>=0.25.0->jupyterlab->jupyter)
  Downloading h11-0.14.0-py3-none-any.whl.metadata (8.2 kB)
Requirement already satisfied: jedi>=0.16 in /usr/local/lib/python3.10/dist-packages (from ipython>=5.0.0->ipykernel->jupyter)
(0.19.1)
Requirement already satisfied: decorator in /usr/local/lib/python3.10/dist-packages (from ipython>=5.0.0->ipykernel->jupyter)
(4.4.2)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.10/dist-packages (from ipython>=5.0.0->ipykernel->jupyter)
(0.7.5)
Requirement already satisfied: backcall in /usr/local/lib/python3.10/dist-packages (from ipython>=5.0.0->ipykernel->jupyter) (0.
2.0)
Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.10/dist-packages (from ipython>=5.0.0->ipykernel->jupyter)
(4.9.0)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.10/dist-packages (from jupyter-client->ipykernel->
jupyter) (2.8.2)
```

```
Requirement already satisfied: platformdirs>=2.5 in /usr/local/lib/python3.10/dist-packages (from jupyter-core->jupyterlab->jupyter) (4.3.6)
Collecting jupyter-client (from jupyter-console->jupyter)
  Downloading jupyter_client-7.4.9-py3-none-any.whl.metadata (8.5 kB)
Collecting jupyter-events>=0.9.0 (from jupyter-server<3,>=2.4.0->jupyterlab->jupyter)
  Downloading jupyter_events-0.10.0-py3-none-any.whl.metadata (5.9 kB)
Collecting jupyter-server-terminals>=0.4.4 (from jupyter-server<3,>=2.4.0->jupyterlab->jupyter)
  Downloading jupyter_server_terminals-0.5.3-py3-none-any.whl.metadata (5.6 kB)
Collecting overrides>=5.0 (from jupyter-server<3,>=2.4.0->jupyterlab->jupyter)
  Downloading overrides-7.7.0-py3-none-any.whl.metadata (5.8 kB)
Requirement already satisfied: websocket-client>=1.7 in /usr/local/lib/python3.10/dist-packages (from jupyter-server<3,>=2.4.0->jupyterlab->jupyter) (1.8.0)
Requirement already satisfied: argon2-cffi-bindings in /usr/local/lib/python3.10/dist-packages (from argon2-cffi->notebook->jupyter) (21.2.0)
Requirement already satisfied: babel>=2.10 in /usr/local/lib/python3.10/dist-packages (from jupyterlab-server<3,>=2.27.1->jupyterlab->jupyter) (2.16.0)
Collecting json5>=0.9.0 (from jupyterlab-server<3,>=2.27.1->jupyterlab->jupyter)
  Downloading json5-0.9.25-py3-none-any.whl.metadata (30 kB)
Requirement already satisfied: jsonschema>=4.18.0 in /usr/local/lib/python3.10/dist-packages (from jupyterlab-server<3,>=2.27.1->jupyterlab->jupyter) (4.23.0)
Requirement already satisfied: requests>=2.31 in /usr/local/lib/python3.10/dist-packages (from jupyterlab-server<3,>=2.27.1->jupyterlab->jupyter) (2.32.3)
Requirement already satisfied: fastjsonschema>=2.15 in /usr/local/lib/python3.10/dist-packages (from nbformat>=5.1->nbconvert->jupyter) (2.20.0)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.10/dist-packages (from prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0->jupyter-console->jupyter) (0.2.13)
Requirement already satisfied: ptyprocess in /usr/local/lib/python3.10/dist-packages (from terminado>=0.8.3->notebook->jupyter) (0.7.0)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-packages (from beautifulsoup4->nbconvert->jupyter) (2.6)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.10/dist-packages (from bleach->nbconvert->jupyter) (1.16.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (from bleach->nbconvert->jupyter) (0.5.1)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from anyio->httpx>=0.25.0->jupyterlab->jupyter) (1.2.2)
Requirement already satisfied: parso<0.9.0,>=0.8.3 in /usr/local/lib/python3.10/dist-packages (from jedi>=0.16->ipython>=5.0.0->ipykernel->jupyter) (0.8.4)
Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=4.18.0->jupyterlab-server<3,>=2.27.1->jupyterlab->jupyter) (24.2.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=4.18.0->jupyterlab-server<3,>=2.27.1->jupyterlab->jupyter) (2023.12.1)
Requirement already satisfied: referencing>=0.28.4 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=4.18.0->jupyterlab-server<3,>=2.27.1->jupyterlab->jupyter) (0.35.1)
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=4.18.0->jupyterlab-server<3,>=2.27.1->jupyterlab->jupyter) (0.20.0)
```

```
Collecting python-json-logger>=2.0.4 (from jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter)
  Downloading python_json_logger-2.0.7-py3-none-any.whl.metadata (6.5 kB)
Requirement already satisfied: pyyaml>=5.3 in /usr/local/lib/python3.10/dist-packages (from jupyter-events>=0.9.0->jupyter-serve
r<3,>=2.4.0->jupyterlab->jupyter) (6.0.2)
Collecting rfc3339-validator (from jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter)
  Downloading rfc3339_validator-0.1.4-py2.py3-none-any.whl.metadata (1.5 kB)
Collecting rfc3986-validator>=0.1.1 (from jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter)
  Downloading rfc3986_validator-0.1.1-py2.py3-none-any.whl.metadata (1.7 kB)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->jupyter
lab-server<3,>=2.27.1->jupyterlab->jupyter) (3.3.2)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->jupyterlab-se
rver<3,>=2.27.1->jupyterlab->jupyter) (2.2.3)
Requirement already satisfied: cffi>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from argon2-cffi-bindings->argon2-cffi->n
otebook->jupyter) (1.17.1)
Requirement already satisfied: pycparser in /usr/local/lib/python3.10/dist-packages (from cffi>=1.0.1->argon2-cffi-bindings->arg
on2-cffi->notebook->jupyter) (2.22)
Collecting fqdn (from jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter)
  Downloading fqdn-1.5.1-py3-none-any.whl.metadata (1.4 kB)
Collecting isoduration (from jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jup
yter)
  Downloading isoduration-20.11.0-py3-none-any.whl.metadata (5.7 kB)
Collecting jsonpointer>1.13 (from jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab
->jupyter)
  Downloading jsonpointer-3.0.0-py2.py3-none-any.whl.metadata (2.3 kB)
Collecting uri-template (from jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->ju
pyter)
  Downloading uri_template-1.3.0-py3-none-any.whl.metadata (8.8 kB)
Requirement already satisfied: webcolors>=24.6.0 in /usr/local/lib/python3.10/dist-packages (from jsonschema[format-nongpl]>=4.1
8.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter) (24.8.0)
Collecting arrow>=0.15.0 (from isoduration->jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->
jupyterlab->jupyter)
  Downloading arrow-1.3.0-py3-none-any.whl.metadata (7.5 kB)
Collecting types-python-dateutil>=2.8.10 (from arrow>=0.15.0->isoduration->jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.
9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter)
  Downloading types_python_dateutil-2.9.0.20240906-py3-none-any.whl.metadata (1.9 kB)
Downloading jupyter-1.1.1-py2.py3-none-any.whl (2.7 kB)
Downloading jupyterlab-4.2.5-py3-none-any.whl (11.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 11.6/11.6 MB 41.2 MB/s eta 0:00:00
Downloading ipykernel-6.29.5-py3-none-any.whl (117 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 117.2/117.2 kB 10.7 MB/s eta 0:00:00
Downloading async_lru-2.0.4-py3-none-any.whl (6.1 kB)
Downloading httpx-0.27.2-py3-none-any.whl (76 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 76.4/76.4 kB 7.5 MB/s eta 0:00:00
Downloading httpcore-1.0.5-py3-none-any.whl (77 kB)
```

```
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 77.9/77.9 kB 3.5 MB/s eta 0:00:00
Downloading jupyter_lsp-2.2.5-py3-none-any.whl (69 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 69.1/69.1 kB 6.4 MB/s eta 0:00:00
Downloading jupyter_server-2.14.2-py3-none-any.whl (383 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 383.6/383.6 kB 29.1 MB/s eta 0:00:00
Downloading jupyter_client-7.4.9-py3-none-any.whl (133 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 133.5/133.5 kB 8.9 MB/s eta 0:00:00
Downloading jupyterlab_server-2.27.3-py3-none-any.whl (59 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 59.7/59.7 kB 5.2 MB/s eta 0:00:00
Downloading json5-0.9.25-py3-none-any.whl (30 kB)
Downloading jupyter_events-0.10.0-py3-none-any.whl (18 kB)
Downloading jupyter_server_terminals-0.5.3-py3-none-any.whl (13 kB)
Downloading overrides-7.7.0-py3-none-any.whl (17 kB)
Downloading h11-0.14.0-py3-none-any.whl (58 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 58.3/58.3 kB 5.4 MB/s eta 0:00:00
Downloading python_json_logger-2.0.7-py3-none-any.whl (8.1 kB)
Downloading rfc3986_validator-0.1.1-py2.py3-none-any.whl (4.2 kB)
Downloading rfc3339_validator-0.1.4-py2.py3-none-any.whl (3.5 kB)
Downloading jsonpointer-3.0.0-py2.py3-none-any.whl (7.6 kB)
Downloading fqdn-1.5.1-py3-none-any.whl (9.1 kB)
Downloading isoduration-20.11.0-py3-none-any.whl (11 kB)
Downloading uri_template-1.3.0-py3-none-any.whl (11 kB)
Downloading arrow-1.3.0-py3-none-any.whl (66 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 66.4/66.4 kB 6.4 MB/s eta 0:00:00
Downloading types_python_dateutil-2.9.0.20240906-py3-none-any.whl (9.7 kB)
Installing collected packages: uri-template, types-python-dateutil, rfc3986-validator, rfc3339-validator, python-json-logger, ov
errides, jsonpointer, json5, h11, fqdn, async-lru, jupyter-server-terminals, jupyter-client, httpcore, arrow, isoduration, ipyke
rnel, httpx, jupyter-events, jupyter-server, jupyterlab-server, jupyter-lsp, jupyterlab, jupyter
  Attempting uninstall: jupyter-client
    Found existing installation: jupyter-client 6.1.12
    Uninstalling jupyter-client-6.1.12:
      Successfully uninstalled jupyter-client-6.1.12
  Attempting uninstall: ipykernel
    Found existing installation: ipykernel 5.5.6
    Uninstalling ipykernel-5.5.6:
      Successfully uninstalled ipykernel-5.5.6
  Attempting uninstall: jupyter-server
    Found existing installation: jupyter-server 1.24.0
    Uninstalling jupyter-server-1.24.0:
      Successfully uninstalled jupyter-server-1.24.0
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the
source of the following dependency conflicts.
google-colab 1.0.0 requires ipykernel==5.5.6, but you have ipykernel 6.29.5 which is incompatible.
Successfully installed arrow-1.3.0 async-lru-2.0.4 fqdn-1.5.1 h11-0.14.0 httpcore-1.0.5 httpx-0.27.2 ipykernel-6.29.5 isoduratio
```

```
n-20.11.0 json5-0.9.25 jsonpointer-3.0.0 jupyter-1.1.1 jupyter-client-7.4.9 jupyter-events-0.10.0 jupyter-lsp-2.2.5 jupyter-serv
er-2.14.2 jupyter-server-terminals-0.5.3 jupyterlab-4.2.5 jupyterlab-server-2.27.3 overrides-7.7.0 python-json-logger-2.0.7 rfc3
339-validator-0.1.4 rfc3986-validator-0.1.1 types-python-dateutil-2.9.0.20240906 uri-template-1.3.0
```

In [ ]:

Out[ ]: 4

In [4]:
```
!pip install torch==2.4.0
```

```
Collecting torch==2.4.0
  Downloading torch-2.4.0-cp310-cp310-manylinux1_x86_64.whl.metadata (26 kB)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch==2.4.0) (3.16.1)
Requirement already satisfied: typing-extensions>=4.8.0 in /usr/local/lib/python3.10/dist-packages (from torch==2.4.0) (4.12.2)
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch==2.4.0) (1.13.3)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch==2.4.0) (3.3)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch==2.4.0) (3.1.4)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from torch==2.4.0) (2024.6.1)
Collecting nvidia-cuda-nvrtc-cu12==12.1.105 (from torch==2.4.0)
  Downloading nvidia_cuda_nvrtc_cu12-12.1.105-py3-none-manylinux1_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cuda-runtime-cu12==12.1.105 (from torch==2.4.0)
  Downloading nvidia_cuda_runtime_cu12-12.1.105-py3-none-manylinux1_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cuda-cupti-cu12==12.1.105 (from torch==2.4.0)
  Downloading nvidia_cuda_cupti_cu12-12.1.105-py3-none-manylinux1_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cudnn-cu12==9.1.0.70 (from torch==2.4.0)
  Downloading nvidia_cudnn_cu12-9.1.0.70-py3-none-manylinux2014_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cublas-cu12==12.1.3.1 (from torch==2.4.0)
  Downloading nvidia_cublas_cu12-12.1.3.1-py3-none-manylinux1_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cufft-cu12==11.0.2.54 (from torch==2.4.0)
  Downloading nvidia_cufft_cu12-11.0.2.54-py3-none-manylinux1_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-curand-cu12==10.3.2.106 (from torch==2.4.0)
  Downloading nvidia_curand_cu12-10.3.2.106-py3-none-manylinux1_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cusolver-cu12==11.4.5.107 (from torch==2.4.0)
  Downloading nvidia_cusolver_cu12-11.4.5.107-py3-none-manylinux1_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cusparse-cu12==12.1.0.106 (from torch==2.4.0)
  Downloading nvidia_cusparse_cu12-12.1.0.106-py3-none-manylinux1_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-nccl-cu12==2.20.5 (from torch==2.4.0)
  Downloading nvidia_nccl_cu12-2.20.5-py3-none-manylinux2014_x86_64.whl.metadata (1.8 kB)
Collecting nvidia-nvtx-cu12==12.1.105 (from torch==2.4.0)
  Downloading nvidia_nvtx_cu12-12.1.105-py3-none-manylinux1_x86_64.whl.metadata (1.7 kB)
Requirement already satisfied: triton==3.0.0 in /usr/local/lib/python3.10/dist-packages (from torch==2.4.0) (3.0.0)
Requirement already satisfied: nvidia-nvjitlink-cu12 in /usr/local/lib/python3.10/dist-packages (from nvidia-cusolver-cu12==11.
4.5.107->torch==2.4.0) (12.6.68)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch==2.4.0) (2.1.5)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy->torch==2.4.0) (1.3.0)
Downloading torch-2.4.0-cp310-cp310-manylinux1_x86_64.whl (797.2 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 797.2/797.2 MB 1.0 MB/s eta 0:00:00
Downloading nvidia_cublas_cu12-12.1.3.1-py3-none-manylinux1_x86_64.whl (410.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 410.6/410.6 MB 1.3 MB/s eta 0:00:00
Downloading nvidia_cuda_cupti_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (14.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 14.1/14.1 MB 32.2 MB/s eta 0:00:00
Downloading nvidia_cuda_nvrtc_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (23.7 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 23.7/23.7 MB 56.6 MB/s eta 0:00:00
Downloading nvidia_cuda_runtime_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (823 kB)
```

```
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 823.6/823.6 kB 47.3 MB/s eta 0:00:00
Downloading nvidia_cudnn_cu12-9.1.0.70-py3-none-manylinux2014_x86_64.whl (664.8 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 664.8/664.8 MB 2.9 MB/s eta 0:00:00
Downloading nvidia_cufft_cu12-11.0.2.54-py3-none-manylinux1_x86_64.whl (121.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 121.6/121.6 MB 7.8 MB/s eta 0:00:00
Downloading nvidia_curand_cu12-10.3.2.106-py3-none-manylinux1_x86_64.whl (56.5 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 56.5/56.5 MB 14.5 MB/s eta 0:00:00
Downloading nvidia_cusolver_cu12-11.4.5.107-py3-none-manylinux1_x86_64.whl (124.2 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 124.2/124.2 MB 7.8 MB/s eta 0:00:00
Downloading nvidia_cusparse_cu12-12.1.0.106-py3-none-manylinux1_x86_64.whl (196.0 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 196.0/196.0 MB 6.3 MB/s eta 0:00:00
Downloading nvidia_nccl_cu12-2.20.5-py3-none-manylinux2014_x86_64.whl (176.2 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 176.2/176.2 MB 6.8 MB/s eta 0:00:00
Downloading nvidia_nvtx_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (99 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 99.1/99.1 kB 9.8 MB/s eta 0:00:00
Installing collected packages: nvidia-nvtx-cu12, nvidia-nccl-cu12, nvidia-cusparse-cu12, nvidia-curand-cu12, nvidia-cufft-cu12,
nvidia-cuda-runtime-cu12, nvidia-cuda-nvrtc-cu12, nvidia-cuda-cupti-cu12, nvidia-cublas-cu12, nvidia-cusolver-cu12, nvidia-cudnn
-cu12, torch
  Attempting uninstall: nvidia-nccl-cu12
    Found existing installation: nvidia-nccl-cu12 2.23.4
    Uninstalling nvidia-nccl-cu12-2.23.4:
      Successfully uninstalled nvidia-nccl-cu12-2.23.4
  Attempting uninstall: nvidia-cusparse-cu12
    Found existing installation: nvidia-cusparse-cu12 12.5.3.3
    Uninstalling nvidia-cusparse-cu12-12.5.3.3:
      Successfully uninstalled nvidia-cusparse-cu12-12.5.3.3
  Attempting uninstall: nvidia-cufft-cu12
    Found existing installation: nvidia-cufft-cu12 11.2.6.59
    Uninstalling nvidia-cufft-cu12-11.2.6.59:
      Successfully uninstalled nvidia-cufft-cu12-11.2.6.59
  Attempting uninstall: nvidia-cuda-runtime-cu12
    Found existing installation: nvidia-cuda-runtime-cu12 12.6.68
    Uninstalling nvidia-cuda-runtime-cu12-12.6.68:
      Successfully uninstalled nvidia-cuda-runtime-cu12-12.6.68
  Attempting uninstall: nvidia-cuda-cupti-cu12
    Found existing installation: nvidia-cuda-cupti-cu12 12.6.68
    Uninstalling nvidia-cuda-cupti-cu12-12.6.68:
      Successfully uninstalled nvidia-cuda-cupti-cu12-12.6.68
  Attempting uninstall: nvidia-cublas-cu12
    Found existing installation: nvidia-cublas-cu12 12.6.1.4
    Uninstalling nvidia-cublas-cu12-12.6.1.4:
      Successfully uninstalled nvidia-cublas-cu12-12.6.1.4
  Attempting uninstall: nvidia-cusolver-cu12
    Found existing installation: nvidia-cusolver-cu12 11.6.4.69
```

```
    Uninstalling nvidia-cusolver-cu12-11.6.4.69:
      Successfully uninstalled nvidia-cusolver-cu12-11.6.4.69
  Attempting uninstall: nvidia-cudnn-cu12
    Found existing installation: nvidia-cudnn-cu12 9.4.0.58
    Uninstalling nvidia-cudnn-cu12-9.4.0.58:
      Successfully uninstalled nvidia-cudnn-cu12-9.4.0.58
  Attempting uninstall: torch
    Found existing installation: torch 2.4.1+cu121
    Uninstalling torch-2.4.1+cu121:
      Successfully uninstalled torch-2.4.1+cu121
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the
source of the following dependency conflicts.
torchaudio 2.4.1+cu121 requires torch==2.4.1, but you have torch 2.4.0 which is incompatible.
torchvision 0.19.1+cu121 requires torch==2.4.1, but you have torch 2.4.0 which is incompatible.
xformers 0.0.28.post1 requires torch==2.4.1, but you have torch 2.4.0 which is incompatible.
Successfully installed nvidia-cublas-cu12-12.1.3.1 nvidia-cuda-cupti-cu12-12.1.105 nvidia-cuda-nvrtc-cu12-12.1.105 nvidia-cuda-r
untime-cu12-12.1.105 nvidia-cudnn-cu12-9.1.0.70 nvidia-cufft-cu12-11.0.2.54 nvidia-curand-cu12-10.3.2.106 nvidia-cusolver-cu12-1
1.4.5.107 nvidia-cusparse-cu12-12.1.0.106 nvidia-nccl-cu12-2.20.5 nvidia-nvtx-cu12-12.1.105 torch-2.4.0
```

In [4]:

2.1 Import Necessary Libraries (3 Marks)

In [5]:
```python
# import FastLanguageModel from unsloth library
from unsloth import FastLanguageModel
# import SFTTrainer from trl
from trl import SFTTrainer
# import TrainingArguments from transformers
from transformers import TrainingArguments
# import evaluate library
import evaluate
# import locale library
import locale
```

```
🦥 Unsloth: Will patch your computer to enable 2x faster free finetuning.
```

In [5]:

In [6]:
```
import pandas as pd
import numpy as np
import datasets
import torch
```

2.2 Dataset Preprocessing for Text Summarization (3 Marks)

(A) Read the Dataset (1 Marks)

(B) Split the Datset (2 Marks)

2.2 (A) Load Dataset

In [9]:
```
sample_reviews_df= pd.read_csv("/content/customer_reviews_dataset.csv")
```

In [10]:
```
sample_reviews_df['dialogue'] = 'customer : ' + sample_reviews_df['review_text'] + '\n' + 'response : ' + sample_reviews_df['resp
```

In [11]:
```
sample_reviews_df['id'] = sample_reviews_df['customer_id']
```

In [12]:
```
sample_reviews_df = sample_reviews_df[['id', 'review_sentiment' ,'dialogue','summary']]
```

In [13]:
```
sample_reviews_df.head()
```

Out[13]:

	id	review_sentiment	dialogue	summary
0	CID041	Positive	customer : I bought this laptop for my son who...	The user purchased a laptop for their son, who...
1	CID011	Negative	customer : I was very disappointed with these ...	The user expressed disappointment with poor so...
2	CID034	Positive	customer : Awesome power bank, it charges my p...	The user praises the power bank for its fast c...
3	CID032	Negative	customer : I bought this phone mainly for its ...	The user expressed disappointment with the pho...
4	CID051	Positive	customer : I love these headphones, they are a...	The user praises the headphones for their exce...

2.2 (B) Split Dataset

In [14]:
```
# Separate positive and negative reviews
positive_reviews =  sample_reviews_df[sample_reviews_df['review_sentiment'] == 'Positive']
negative_reviews =  sample_reviews_df[sample_reviews_df['review_sentiment'] == 'Negative']

# Sample 2 positive and 2 negative reviews for gold examples
positive_gold_examples = positive_reviews.sample(2, random_state=40)
negative_gold_examples = negative_reviews.sample(2, random_state=40)

# Concatenate positive and negative gold examples
sample_reviews_gold_examples_df =  pd.concat([positive_gold_examples, negative_gold_examples])


# Create the training set by excluding gold examples
sample_reviews_examples_df =  sample_reviews_df.drop(sample_reviews_gold_examples_df.index)

# Print the shapes of the datasets
print("Training Set Shape:", sample_reviews_examples_df.shape)
print("Gold Examples Shape:", sample_reviews_gold_examples_df.shape)
```

```
Training Set Shape: (26, 4)
Gold Examples Shape: (4, 4)
```

2.3 Model Setup for Fine-tuning (5 Marks)

(A) Load Model Name (1 Marks)

(B) Create Examples (2 Marks)

(C) Initialize Model (2 Marks)

2.3 (A) Load llama-2-7b-bnb-4bit model Name from unsloth as unsloth/model name

In [15]:
```
model, tokenizer = FastLanguageModel.from_pretrained(
    model_name="unsloth/llama-2-7b-bnb-4bit",
    max_seq_length=2048,
    dtype=None, # None for auto detection. Float16 for Tesla T4, V100, Bfloat16 for Ampere+
    load_in_4bit=True # Use 4bit quantization to reduce memory usage.
)
```

```
==((====))==  Unsloth 2024.9.post2: Fast Llama patching. Transformers = 4.44.2.
   \\   /|    GPU: Tesla T4. Max memory: 14.748 GB. Platform = Linux.
O^O/ \_/ \    Pytorch: 2.4.0+cu121. CUDA = 7.5. CUDA Toolkit = 12.1.
\        /    Bfloat16 = FALSE. FA [Xformers = 0.0.28.post1. FA2 = False]
 "-____-"     Free Apache license: http://github.com/unslothai/unsloth
Unsloth: Fast downloading is enabled - ignore downloading bars which are red colored!
model.safetensors:   0%|          | 0.00/3.87G [00:00<?, ?B/s]
generation_config.json:   0%|          | 0.00/183 [00:00<?, ?B/s]
tokenizer_config.json:   0%|          | 0.00/948 [00:00<?, ?B/s]
tokenizer.model:   0%|          | 0.00/500k [00:00<?, ?B/s]
special_tokens_map.json:   0%|          | 0.00/438 [00:00<?, ?B/s]
tokenizer.json:   0%|          | 0.00/1.84M [00:00<?, ?B/s]
```

In [ ]:
```
#from google.colab import output
#output.enable_custom_widget_manager()
```

Support for third party widgets will remain active for the duration of the session. To disable support:

In [ ]:
```
#from google.colab import output
#output.disable_custom_widget_manager()
```

In [107…

In [16]:
```
model
```

Out[16]:
```
LlamaForCausalLM(
  (model): LlamaModel(
    (embed_tokens): Embedding(32000, 4096)
    (layers): ModuleList(
      (0-31): 32 x LlamaDecoderLayer(
        (self_attn): LlamaAttention(
          (q_proj): Linear4bit(in_features=4096, out_features=4096, bias=False)
          (k_proj): Linear4bit(in_features=4096, out_features=4096, bias=False)
          (v_proj): Linear4bit(in_features=4096, out_features=4096, bias=False)
          (o_proj): Linear4bit(in_features=4096, out_features=4096, bias=False)
          (rotary_emb): LlamaRotaryEmbedding()
        )
        (mlp): LlamaMLP(
          (gate_proj): Linear4bit(in_features=4096, out_features=11008, bias=False)
          (up_proj): Linear4bit(in_features=4096, out_features=11008, bias=False)
          (down_proj): Linear4bit(in_features=11008, out_features=4096, bias=False)
          (act_fn): SiLU()
        )
        (input_layernorm): LlamaRMSNorm((4096,), eps=1e-05)
        (post_attention_layernorm): LlamaRMSNorm((4096,), eps=1e-05)
      )
    )
    (norm): LlamaRMSNorm((4096,), eps=1e-05)
    (rotary_emb): LlamaRotaryEmbedding()
  )
  (lm_head): Linear(in_features=4096, out_features=32000, bias=False)
)
```

In [17]:
```
tokenizer
```

Out[17]:
```
LlamaTokenizerFast(name_or_path='unsloth/llama-2-7b-bnb-4bit', vocab_size=32000, model_max_length=4096, is_fast=True, padding_si
de='left', truncation_side='right', special_tokens={'bos_token': '<s>', 'eos_token': '</s>', 'unk_token': '<unk>', 'pad_token':
'<unk>'}, clean_up_tokenization_spaces=False),  added_tokens_decoder={
        0: AddedToken("<unk>", rstrip=False, lstrip=False, single_word=False, normalized=False, special=True),
        1: AddedToken("<s>", rstrip=False, lstrip=False, single_word=False, normalized=False, special=True),
        2: AddedToken("</s>", rstrip=False, lstrip=False, single_word=False, normalized=False, special=True),
}
```

In [18]:
```
EOS_TOKEN = tokenizer.eos_token
```

In [19]:
```
model = FastLanguageModel.get_peft_model(
    model,
    r=16,
    target_modules = ["q_proj", "k_proj", "v_proj", "o_proj",
```

```
                      "gate_proj", "up_proj", "down_proj",],
    lora_alpha=16,
    lora_dropout=0, # Supports any, but = 0 is optimized
    bias="none",    # Supports any, but = "none" is optimized
    use_gradient_checkpointing=True,
    random_state=42,
    use_rslora=False,
    loftq_config=None
)
```

```
Unsloth 2024.9.post2 patched 32 layers with 32 QKV layers, 32 O layers and 32 MLP layers.
```

In [ ]:

In [20]:
```
alpaca_prompt = """Below is an instruction that describes a task, paired with an input that provides further context. Write a res

### Instruction:
{}

### Input:
{}

### Response:
{}"""
```

In [ ]:

In [21]:
```
def create_examples_with_seed(dataset, n=2, random_seed=None):
    """
    Return two DataFrames with randomized examples of size 2n with two classes.
    Create subsets of each class, choose random samples from the subsets,
    merge and randomize the order of samples in the merged list.
    Each run of this function creates a different random sample of examples
    chosen from the training data.

    Args:
        dataset (DataFrame): A DataFrame with examples (text + label)
        n (int): number of examples of each class to be selected
        random_seed (int): seed for reproducibility (default is None)

    Output:
        few_shot_examples_df (DataFrame): A DataFrame with examples in random order
        new_df (DataFrame): A new DataFrame excluding selected examples
```

```
    """

    positive_reviews = (dataset.review_sentiment == 'Positive')
    negative_reviews = (dataset.review_sentiment == 'Negative')
    columns_to_select = ['id', 'review_sentiment' ,'dialogue','summary']

    # Set a fixed random seed for reproducibility
    np.random.seed(random_seed)

    positive_examples = dataset.loc[positive_reviews, columns_to_select].sample(n)
    negative_examples = dataset.loc[negative_reviews, columns_to_select].sample(n)

    few_shot_examples_df = pd.concat([positive_examples, negative_examples])
    # sampling without replacement is equivalent to random shuffling
    few_shot_examples_df = few_shot_examples_df.sample(2 * n, replace=False)

    # Create a new DataFrame excluding selected examples
    new_df = dataset.drop(index=few_shot_examples_df.index)

    return few_shot_examples_df, new_df
```

In [ ]:

2.3 (B) Create Examples by mentioning the seed value used earlier and n=2

In [22]:
```
sample_reviews_train_examples_df , sample_reviews_validation_examples_df = create_examples_with_seed(sample_reviews_df, n=2, ranc
```

In [ ]:

In [23]:
```
training_dataset = datasets.Dataset.from_pandas(sample_reviews_train_examples_df)
validation_dataset = datasets.Dataset.from_pandas(sample_reviews_validation_examples_df)
```

In [ ]:

In [24]:
```
def prompt_formatter(example, prompt_template):
    instruction='Summarize the following dialogue'
    dialogue=example["dialogue"]
    summary=example["summary"]

    formatted_prompt = prompt_template.format(instruction, dialogue, summary)
```

```
    return {'formatted_prompt': formatted_prompt}
```

In [25]:
```
formatted_training_dataset = training_dataset.map(
    prompt_formatter,
    fn_kwargs={'prompt_template': alpaca_prompt}
)
```

```
Map:   0%|          | 0/4 [00:00<?, ? examples/s]
```

In [ ]:
```
```

In [26]:
```
formatted_validation_dataset = validation_dataset.map(
    prompt_formatter,
    fn_kwargs={'prompt_template': alpaca_prompt}
)
```

```
Map:   0%|          | 0/26 [00:00<?, ? examples/s]
```

In [27]:
```
# Optional: Display the first few formatted prompts for verification
print(formatted_validation_dataset['formatted_prompt'][:5])  # Display first 5 formatted prompts
```

["Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appro
priately completes the request.\n\n### Instruction:\nSummarize the following dialogue\n\n### Input:\ncustomer : I bought this la
ptop for my son who is studying engineering. He is very happy with it. It has a good battery life, fast performance, and a sleek
design. The keyboard is comfortable and the screen is bright. The laptop came with a one-year warranty and a free antivirus soft
ware. I think it is a great value for money.\nresponse : It's fantastic to hear that the laptop you purchased for your son has m
et his needs and expectations, especially in his engineering studies. A good battery life, fast performance, and sleek design ar
e essential for a student's productivity. The comfortable keyboard and bright screen further enhance the usability of the lapto
p. If you ever encounter any issues or have questions about the laptop, please feel free to reach out for support. We're here to
ensure that your experience continues to be positive. Thank you for choosing our product and taking the time to share your satis
faction!\n\n\n### Response:\nThe user purchased a laptop for their son, who is studying engineering. They are satisfied with its
battery life, fast performance, sleek design, comfortable keyboard, and bright screen, and its one-year warranty.", "Below is an
instruction that describes a task, paired with an input that provides further context. Write a response that appropriately compl
etes the request.\n\n### Instruction:\nSummarize the following dialogue\n\n### Input:\ncustomer : I was very disappointed with t
hese headphones. The sound quality is very poor, the bass is weak, and the treble is harsh. The headphones are also very uncomfo
rtable to wear, they hurt my ears after a few minutes. The Bluetooth connection is also very unstable, it keeps disconnecting an
d reconnecting. Charging takes quite long time. I regret buying these headphones, they are a waste of money.\nresponse : I'm tru
ly sorry to hear about your disappointing experience with our headphones. Your feedback is valuable, and we aim to address these
issues promptly. Regarding the sound quality, we're looking into the balance of bass and treble to ensure a more enjoyable liste
ning experience. For the discomfort and Bluetooth stability, we are exploring ergonomic improvements and firmware updates. Final
ly, we're working on enhancing the charging efficiency. Please contact our customer service for further assistance or potential
remedies. Your satisfaction is our top priority.\n\n\n### Response:\nThe user expressed disappointment with poor sound quality,
weak bass, harsh treble, discomfort, unstable Bluetooth connection, and long charging time. The company is working on improving
sound balance, ergonomics, and charging efficiency.", "Below is an instruction that describes a task, paired with an input that
provides further context. Write a response that appropriately completes the request.\n\n### Instruction:\nSummarize the followin
g dialogue\n\n### Input:\ncustomer : Awesome power bank, it charges my phone very fast and lasts for a long time. It is also ver
y compact and lightweight, easy to carry around. It has a LED indicator that shows the battery level and a dual USB port that ca
n charge two devices at the same time. It also has a low power mode that can charge small devices like earphones and smartwatche
s. I am very satisfied with this product.\nresponse : Thank you for your positive review of our power bank! We're thrilled to he
ar that its fast charging capability, long-lasting power, compact design, and dual USB ports meet your needs effectively. It's g
reat to know that the LED indicator and low power mode for smaller devices add to your satisfaction. Your feedback is highly app
reciated, and we're committed to maintaining this level of quality and convenience in our products. If you need any further assi
stance or information, please feel free to reach out to us.\n\n\n### Response:\nThe user praises the power bank for its fast cha
rging, long-lasting power, compact design, LED indicator, dual USB ports, and low power mode for small devices, expressing satis
faction with its quality and convenience.", "Below is an instruction that describes a task, paired with an input that provides f
urther context. Write a response that appropriately completes the request.\n\n### Instruction:\nSummarize the following dialogue
\n\n### Input:\ncustomer : I bought this phone mainly for its much-hyped camera, but I was very disappointed with the results. T
he pictures are blurry, grainy, and overexposed. The zoom is also very bad, it makes the pictures look pixelated and distorted.
The night mode is also useless, it makes the pictures look dark and noisy. The video quality is also very poor, it lags and stut
ters. Battery is also not very good. Only good thing is display. I expected much better from a flagship phone.\nresponse : I'm t
ruly sorry to hear that the camera performance of your phone did not meet your expectations, especially considering it's a key f
eature you were looking forward to. We take feedback about picture quality, zoom functionality, night mode, and video performanc
e very seriously. I'll relay your concerns to our development team for further improvement. In the meantime, for the issues with
blurriness, graininess, and battery performance, our customer service team would be glad to assist with troubleshooting or poten
tial solutions. We're committed to ensuring your satisfaction and improving our flagship offerings. Please don't hesitate to rea

ch out for further assistance.\n\n\n### Response:\nThe user expressed disappointment with the phone's camera performance, statin
g blurry, grainy, overexposed pictures, poor zoom, distorted night mode, lags, stutters, and poor video quality. The phone's cus
tomer service team is working on improving these issues.", "Below is an instruction that describes a task, paired with an input
that provides further context. Write a response that appropriately completes the request.\n\n### Instruction:\nSummarize the fol
lowing dialogue\n\n### Input:\ncustomer : I love these headphones, they are amazing. The sound quality is very good, the bass is
deep, and the treble is clear. The headphones are also very comfortable to wear, they fit snugly and do not fall off. The Blueto
oth connection is also very stable, it does not drop or lag. The battery life is also very long, it lasts for more than 10 hour
s. Charging speed is also good . The headphones also have a built-in microphone that works well for calls and voice commands. I
think these headphones are worth every penny.\nresponse : Thank you for your wonderful feedback on our headphones! We're delight
ed to hear that you're enjoying the sound quality, comfort, stable Bluetooth connection, long battery life, and the efficiency o
f the charging speed. It's great to know that the built-in microphone is serving you well for calls and voice commands. Your sat
isfaction is our goal, and we're thrilled that you find the headphones to be a valuable investment. If you ever need assistance
or have any questions, our customer support team is always here to help. Thank you for choosing our product!\n\n\n### Respons
e:\nThe user praises the headphones for their excellent sound quality, comfort, stable Bluetooth connection, long battery life,
and efficient charging speed. They also mention the built-in microphone for calls and voice commands."]

2.3 (C) Initialize Model Parameters

In [ ]:

In [28]:
```
trainer = SFTTrainer(
    model=model,
    tokenizer=tokenizer,
    train_dataset=formatted_training_dataset,
    eval_dataset=formatted_validation_dataset,
    dataset_text_field = "formatted_prompt",
    max_seq_length=2048,
    dataset_num_proc=2,
    packing=False, # Increases efficiency for short sequences.
    args = TrainingArguments(
        per_device_train_batch_size=2,
        per_device_eval_batch_size=2,
        gradient_accumulation_steps=4,
        warmup_steps=5,
        max_steps=50,
        learning_rate=2e-4,
        fp16=not torch.cuda.is_bf16_supported(),
        bf16=torch.cuda.is_bf16_supported(),
        logging_steps=1,
        optim="adamw_8bit",
        weight_decay=0.01,
        lr_scheduler_type="linear",
        seed=42,
```

```
        output_dir="outputs"
    )
)
```

```
Map (num_proc=2):   0%|          | 0/4 [00:00<?, ? examples/s]
Map (num_proc=2):   0%|          | 0/26 [00:00<?, ? examples/s]
```

```
max_steps is given, it will override any value given in num_train_epochs
```

Start of Training

2.4 Train and Save the Model (5 Marks)

(A) Train the Model (3 Marks)

(B) Save the Model (2 Marks)

In [29]:
```
gpu_stats = torch.cuda.get_device_properties(0)
start_gpu_memory = round(torch.cuda.max_memory_reserved() / 1024 / 1024 / 1024, 3)
max_memory = round(gpu_stats.total_memory / 1024 / 1024 / 1024, 3)
print(f"GPU = {gpu_stats.name}. Max memory = {max_memory} GB.")
print(f"{start_gpu_memory} GB of memory reserved.")
```

```
GPU = Tesla T4. Max memory = 14.748 GB.
3.826 GB of memory reserved.
```

2.4 (A) Train Model

In [30]:
```
trainer.train()
```

```
==((====))==  Unsloth - 2x faster free finetuning | Num GPUs = 1
   \\   /|    Num examples = 4 | Num Epochs = 50
O^O/ \_/ \    Batch size per device = 2 | Gradient Accumulation steps = 4
\        /    Total batch size = 8 | Total steps = 50
 "-____-"     Number of trainable parameters = 39,976,960
/usr/local/lib/python3.10/dist-packages/torch/utils/checkpoint.py:295: FutureWarning: `torch.cpu.amp.autocast(args...)` is depre
cated. Please use `torch.amp.autocast('cpu', args...)` instead.
  with torch.enable_grad(), device_autocast_ctx, torch.cpu.amp.autocast(**ctx.cpu_autocast_kwargs):  # type: ignore[attr-define
d]
/usr/local/lib/python3.10/dist-packages/torch/utils/checkpoint.py:295: FutureWarning: `torch.cpu.amp.autocast(args...)` is depre
cated. Please use `torch.amp.autocast('cpu', args...)` instead.
  with torch.enable_grad(), device_autocast_ctx, torch.cpu.amp.autocast(**ctx.cpu_autocast_kwargs):  # type: ignore[attr-define
d]
```

[50/50 03:40, Epoch 50/50]

Step	Training Loss
1	0.882400
2	0.882400
3	0.878200
4	0.855300
5	0.814600
6	0.759200
7	0.686600
8	0.617100
9	0.551500
10	0.487000
11	0.420500
12	0.356300
13	0.302100
14	0.253900
15	0.212200
16	0.174200
17	0.137200
18	0.106000
19	0.083500
20	0.066100
21	0.052000
22	0.044200
23	0.037100

Step	Training Loss
24	0.031500
25	0.027500
26	0.024600
27	0.021700
28	0.019400
29	0.017000
30	0.014200
31	0.011900
32	0.012400
33	0.010200
34	0.008200
35	0.007700
36	0.006700
37	0.005100
38	0.003900
39	0.003300
40	0.003000
41	0.002800
42	0.002700
43	0.002600
44	0.002500
45	0.002500
46	0.002500
47	0.002500

Step	Training Loss
48	0.002500
49	0.002400
50	0.002400

Out[30]:
```
TrainOutput(global_step=50, training_loss=0.1982699571084231, metrics={'train_runtime': 231.8864, 'train_samples_per_second': 1.
725, 'train_steps_per_second': 0.216, 'total_flos': 2968081815748608.0, 'train_loss': 0.1982699571084231, 'epoch': 50.0})
```

In [ ]:
```
```

In [31]:
```
trainer_stats = trainer.state.log_history
```

In [32]:
```
if trainer_stats:
    print("Last Training Stats:", trainer_stats[-1])
else:
    print("No training stats available.")
```

```
Last Training Stats: {'train_runtime': 231.8864, 'train_samples_per_second': 1.725, 'train_steps_per_second': 0.216, 'total_flo
s': 2968081815748608.0, 'train_loss': 0.1982699571084231, 'epoch': 50.0, 'step': 50}
```

In [33]:
```
trainer_stats
```

Out[33]:

```
[{'loss': 0.8824,
  'grad_norm': 0.1277453601360321,
  'learning_rate': 4e-05,
  'epoch': 1.0,
  'step': 1},
 {'loss': 0.8824,
  'grad_norm': 0.1277453601360321,
  'learning_rate': 8e-05,
  'epoch': 2.0,
  'step': 2},
 {'loss': 0.8782,
  'grad_norm': 0.12925252318382263,
  'learning_rate': 0.00012,
  'epoch': 3.0,
  'step': 3},
 {'loss': 0.8553,
  'grad_norm': 0.1490129977464676,
  'learning_rate': 0.00016,
  'epoch': 4.0,
  'step': 4},
 {'loss': 0.8146,
  'grad_norm': 0.17730776965618134,
  'learning_rate': 0.0002,
  'epoch': 5.0,
  'step': 5},
 {'loss': 0.7592,
  'grad_norm': 0.21535995602607727,
  'learning_rate': 0.00019555555555555556,
  'epoch': 6.0,
  'step': 6},
 {'loss': 0.6866,
  'grad_norm': 0.23042012751102448,
  'learning_rate': 0.00019111111111111114,
  'epoch': 7.0,
  'step': 7},
 {'loss': 0.6171,
  'grad_norm': 0.24868714809417725,
  'learning_rate': 0.00018666666666666667,
  'epoch': 8.0,
  'step': 8},
 {'loss': 0.5515,
  'grad_norm': 0.25827717781066895,
  'learning_rate': 0.00018222222222222224,
  'epoch': 9.0,
```

```
 'step': 9},
{'loss': 0.487,
 'grad_norm': 0.25949493050575256,
 'learning_rate': 0.00017777777777777779,
 'epoch': 10.0,
 'step': 10},
{'loss': 0.4205,
 'grad_norm': 0.27096027135849,
 'learning_rate': 0.00017333333333333334,
 'epoch': 11.0,
 'step': 11},
{'loss': 0.3563,
 'grad_norm': 0.2497715801000595,
 'learning_rate': 0.00016888888888888889,
 'epoch': 12.0,
 'step': 12},
{'loss': 0.3021,
 'grad_norm': 0.20092365145683289,
 'learning_rate': 0.00016444444444444444,
 'epoch': 13.0,
 'step': 13},
{'loss': 0.2539,
 'grad_norm': 0.18807664513587952,
 'learning_rate': 0.00016,
 'epoch': 14.0,
 'step': 14},
{'loss': 0.2122,
 'grad_norm': 0.1983853280544281,
 'learning_rate': 0.00015555555555555556,
 'epoch': 15.0,
 'step': 15},
{'loss': 0.1742,
 'grad_norm': 0.21318937838077545,
 'learning_rate': 0.00015111111111111111,
 'epoch': 16.0,
 'step': 16},
{'loss': 0.1372,
 'grad_norm': 0.20242469012737274,
 'learning_rate': 0.00014666666666666666,
 'epoch': 17.0,
 'step': 17},
{'loss': 0.106,
 'grad_norm': 0.18580399453639984,
 'learning_rate': 0.00014222222222222224,
```

```
 'epoch': 18.0,
 'step': 18},
{'loss': 0.0835,
 'grad_norm': 0.1858706772327423,
 'learning_rate': 0.0001377777777777778,
 'epoch': 19.0,
 'step': 19},
{'loss': 0.0661,
 'grad_norm': 0.16316865384578705,
 'learning_rate': 0.00013333333333333334,
 'epoch': 20.0,
 'step': 20},
{'loss': 0.052,
 'grad_norm': 0.13004058599472046,
 'learning_rate': 0.00012888888888888892,
 'epoch': 21.0,
 'step': 21},
{'loss': 0.0442,
 'grad_norm': 0.13826869428157806,
 'learning_rate': 0.00012444444444444444,
 'epoch': 22.0,
 'step': 22},
{'loss': 0.0371,
 'grad_norm': 0.12558192014694214,
 'learning_rate': 0.00012,
 'epoch': 23.0,
 'step': 23},
{'loss': 0.0315,
 'grad_norm': 0.11546556651592255,
 'learning_rate': 0.00011555555555555555,
 'epoch': 24.0,
 'step': 24},
{'loss': 0.0275,
 'grad_norm': 0.10904467105865479,
 'learning_rate': 0.00011111111111111112,
 'epoch': 25.0,
 'step': 25},
{'loss': 0.0246,
 'grad_norm': 0.11422253400087357,
 'learning_rate': 0.00010666666666666667,
 'epoch': 26.0,
 'step': 26},
{'loss': 0.0217,
 'grad_norm': 0.1329180747270584,
```

```
 'learning_rate': 0.00010222222222222222,
 'epoch': 27.0,
 'step': 27},
{'loss': 0.0194,
 'grad_norm': 0.15195374190807343,
 'learning_rate': 9.777777777777778e-05,
 'epoch': 28.0,
 'step': 28},
{'loss': 0.017,
 'grad_norm': 0.1974887102842331,
 'learning_rate': 9.333333333333334e-05,
 'epoch': 29.0,
 'step': 29},
{'loss': 0.0142,
 'grad_norm': 0.25069230794906616,
 'learning_rate': 8.888888888888889e-05,
 'epoch': 30.0,
 'step': 30},
{'loss': 0.0119,
 'grad_norm': 0.1460523158311844,
 'learning_rate': 8.444444444444444e-05,
 'epoch': 31.0,
 'step': 31},
{'loss': 0.0124,
 'grad_norm': 0.38724708557128906,
 'learning_rate': 8e-05,
 'epoch': 32.0,
 'step': 32},
{'loss': 0.0102,
 'grad_norm': 0.2666332423686981,
 'learning_rate': 7.555555555555556e-05,
 'epoch': 33.0,
 'step': 33},
{'loss': 0.0082,
 'grad_norm': 0.08408526331186295,
 'learning_rate': 7.111111111111112e-05,
 'epoch': 34.0,
 'step': 34},
{'loss': 0.0077,
 'grad_norm': 0.31439104676246643,
 'learning_rate': 6.666666666666667e-05,
 'epoch': 35.0,
 'step': 35},
{'loss': 0.0067,
```

```
 'grad_norm': 0.33714812994003296,
 'learning_rate': 6.222222222222222e-05,
 'epoch': 36.0,
 'step': 36},
{'loss': 0.0051,
 'grad_norm': 0.2271455079317093,
 'learning_rate': 5.7777777777777776e-05,
 'epoch': 37.0,
 'step': 37},
{'loss': 0.0039,
 'grad_norm': 0.07294784486293793,
 'learning_rate': 5.333333333333333e-05,
 'epoch': 38.0,
 'step': 38},
{'loss': 0.0033,
 'grad_norm': 0.04371979460120201,
 'learning_rate': 4.888888888888889e-05,
 'epoch': 39.0,
 'step': 39},
{'loss': 0.003,
 'grad_norm': 0.02995261922478676,
 'learning_rate': 4.4444444444444447e-05,
 'epoch': 40.0,
 'step': 40},
{'loss': 0.0028,
 'grad_norm': 0.027055513113737106,
 'learning_rate': 4e-05,
 'epoch': 41.0,
 'step': 41},
{'loss': 0.0027,
 'grad_norm': 0.020985383540391922,
 'learning_rate': 3.555555555555556e-05,
 'epoch': 42.0,
 'step': 42},
{'loss': 0.0026,
 'grad_norm': 0.015078885480761528,
 'learning_rate': 3.111111111111111e-05,
 'epoch': 43.0,
 'step': 43},
{'loss': 0.0025,
 'grad_norm': 0.012185768224298954,
 'learning_rate': 2.6666666666666667e-05,
 'epoch': 44.0,
 'step': 44},
```

```
{'loss': 0.0025,
 'grad_norm': 0.00856876652687788,
 'learning_rate': 2.2222222222222223e-05,
 'epoch': 45.0,
 'step': 45},
{'loss': 0.0025,
 'grad_norm': 0.008195080794394016,
 'learning_rate': 1.777777777777778e-05,
 'epoch': 46.0,
 'step': 46},
{'loss': 0.0025,
 'grad_norm': 0.00677483668550849,
 'learning_rate': 1.3333333333333333e-05,
 'epoch': 47.0,
 'step': 47},
{'loss': 0.0025,
 'grad_norm': 0.007244057022035122,
 'learning_rate': 8.88888888888889e-06,
 'epoch': 48.0,
 'step': 48},
{'loss': 0.0024,
 'grad_norm': 0.00708611449226737,
 'learning_rate': 4.444444444444445e-06,
 'epoch': 49.0,
 'step': 49},
{'loss': 0.0024,
 'grad_norm': 0.010318215005099773,
 'learning_rate': 0.0,
 'epoch': 50.0,
 'step': 50},
{'train_runtime': 231.8864,
 'train_samples_per_second': 1.725,
 'train_steps_per_second': 0.216,
 'total_flos': 2968081815748608.0,
 'train_loss': 0.1982699571084231,
 'epoch': 50.0,
 'step': 50}]
```

Inference

In [34]:
```
test_dataset = datasets.Dataset.from_pandas(sample_reviews_gold_examples_df)
```

In [35]:
```
test_dataset[0]
```

Out[35]:
```
{'id': 'CID041',
 'review_sentiment': 'Positive',
 'dialogue': "customer : I bought this laptop for my son who is studying engineering. He is very happy with it. It has a good ba
ttery life, fast performance, and a sleek design. The keyboard is comfortable and the screen is bright. The laptop came with a o
ne-year warranty and a free antivirus software. I think it is a great value for money.\nresponse : It's fantastic to hear that t
he laptop you purchased for your son has met his needs and expectations, especially in his engineering studies. A good battery l
ife, fast performance, and sleek design are essential for a student's productivity. The comfortable keyboard and bright screen f
urther enhance the usability of the laptop. If you ever encounter any issues or have questions about the laptop, please feel fre
e to reach out for support. We're here to ensure that your experience continues to be positive. Thank you for choosing our produ
ct and taking the time to share your satisfaction!\n",
 'summary': 'The user purchased a laptop for their son, who is studying engineering. They are satisfied with its battery life, f
ast performance, sleek design, comfortable keyboard, and bright screen, and its one-year warranty.',
 '__index_level_0__': 0}
```

In [36]:
```
instruction = "Summarize the following dialogue"
test_dialogue = test_dataset[0]['dialogue']
test_summary = test_dataset[0]['summary']
```

In [37]:
```
FastLanguageModel.for_inference(model)
```

Out[37]:

```
PeftModelForCausalLM(
  (base_model): LoraModel(
    (model): LlamaForCausalLM(
      (model): LlamaModel(
        (embed_tokens): Embedding(32000, 4096)
        (layers): ModuleList(
          (0-31): 32 x LlamaDecoderLayer(
            (self_attn): LlamaAttention(
              (q_proj): lora.Linear4bit(
                (base_layer): Linear4bit(in_features=4096, out_features=4096, bias=False)
                (lora_dropout): ModuleDict(
                  (default): Identity()
                )
                (lora_A): ModuleDict(
                  (default): Linear(in_features=4096, out_features=16, bias=False)
                )
                (lora_B): ModuleDict(
                  (default): Linear(in_features=16, out_features=4096, bias=False)
                )
                (lora_embedding_A): ParameterDict()
                (lora_embedding_B): ParameterDict()
                (lora_magnitude_vector): ModuleDict()
              )
              (k_proj): lora.Linear4bit(
                (base_layer): Linear4bit(in_features=4096, out_features=4096, bias=False)
                (lora_dropout): ModuleDict(
                  (default): Identity()
                )
                (lora_A): ModuleDict(
                  (default): Linear(in_features=4096, out_features=16, bias=False)
                )
                (lora_B): ModuleDict(
                  (default): Linear(in_features=16, out_features=4096, bias=False)
                )
                (lora_embedding_A): ParameterDict()
                (lora_embedding_B): ParameterDict()
                (lora_magnitude_vector): ModuleDict()
              )
              (v_proj): lora.Linear4bit(
                (base_layer): Linear4bit(in_features=4096, out_features=4096, bias=False)
                (lora_dropout): ModuleDict(
                  (default): Identity()
                )
                (lora_A): ModuleDict(
```

```
      (default): Linear(in_features=4096, out_features=16, bias=False)
    )
    (lora_B): ModuleDict(
      (default): Linear(in_features=16, out_features=4096, bias=False)
    )
    (lora_embedding_A): ParameterDict()
    (lora_embedding_B): ParameterDict()
    (lora_magnitude_vector): ModuleDict()
  )
  (o_proj): lora.Linear4bit(
    (base_layer): Linear4bit(in_features=4096, out_features=4096, bias=False)
    (lora_dropout): ModuleDict(
      (default): Identity()
    )
    (lora_A): ModuleDict(
      (default): Linear(in_features=4096, out_features=16, bias=False)
    )
    (lora_B): ModuleDict(
      (default): Linear(in_features=16, out_features=4096, bias=False)
    )
    (lora_embedding_A): ParameterDict()
    (lora_embedding_B): ParameterDict()
    (lora_magnitude_vector): ModuleDict()
  )
  (rotary_emb): LlamaRotaryEmbedding()
)
(mlp): LlamaMLP(
  (gate_proj): lora.Linear4bit(
    (base_layer): Linear4bit(in_features=4096, out_features=11008, bias=False)
    (lora_dropout): ModuleDict(
      (default): Identity()
    )
    (lora_A): ModuleDict(
      (default): Linear(in_features=4096, out_features=16, bias=False)
    )
    (lora_B): ModuleDict(
      (default): Linear(in_features=16, out_features=11008, bias=False)
    )
    (lora_embedding_A): ParameterDict()
    (lora_embedding_B): ParameterDict()
    (lora_magnitude_vector): ModuleDict()
  )
  (up_proj): lora.Linear4bit(
    (base_layer): Linear4bit(in_features=4096, out_features=11008, bias=False)
```

```
                (lora_dropout): ModuleDict(
                  (default): Identity()
                )
                (lora_A): ModuleDict(
                  (default): Linear(in_features=4096, out_features=16, bias=False)
                )
                (lora_B): ModuleDict(
                  (default): Linear(in_features=16, out_features=11008, bias=False)
                )
                (lora_embedding_A): ParameterDict()
                (lora_embedding_B): ParameterDict()
                (lora_magnitude_vector): ModuleDict()
              )
              (down_proj): lora.Linear4bit(
                (base_layer): Linear4bit(in_features=11008, out_features=4096, bias=False)
                (lora_dropout): ModuleDict(
                  (default): Identity()
                )
                (lora_A): ModuleDict(
                  (default): Linear(in_features=11008, out_features=16, bias=False)
                )
                (lora_B): ModuleDict(
                  (default): Linear(in_features=16, out_features=4096, bias=False)
                )
                (lora_embedding_A): ParameterDict()
                (lora_embedding_B): ParameterDict()
                (lora_magnitude_vector): ModuleDict()
              )
              (act_fn): SiLU()
            )
            (input_layernorm): LlamaRMSNorm((4096,), eps=1e-05)
            (post_attention_layernorm): LlamaRMSNorm((4096,), eps=1e-05)
          )
        )
        (norm): LlamaRMSNorm((4096,), eps=1e-05)
        (rotary_emb): LlamaRotaryEmbedding()
      )
      (lm_head): Linear(in_features=4096, out_features=32000, bias=False)
    )
  )
)
```

In [38]:
```
inputs = tokenizer(
[
```

```
    alpaca_prompt.format(
        instruction,
        test_dialogue,
        "", # leave output blank for generation
    )
], return_tensors="pt").to("cuda")
```

In [39]:
```
outputs = model.generate(
    **inputs,
    max_new_tokens=128,
    use_cache=True,
    do_sample=True,
    temperature=0.2
)
```

In [40]:
```
print(tokenizer.batch_decode(outputs)[0])
```

```
<s> Below is an instruction that describes a task, paired with an input that provides further context. Write a response that app
ropriately completes the request.

### Instruction:
Summarize the following dialogue

### Input:
customer : I bought this laptop for my son who is studying engineering. He is very happy with it. It has a good battery life, fa
st performance, and a sleek design. The keyboard is comfortable and the screen is bright. The laptop came with a one-year warran
ty and a free antivirus software. I think it is a great value for money.
response : It's fantastic to hear that the laptop you purchased for your son has met his needs and expectations, especially in h
is engineering studies. A good battery life, fast performance, and sleek design are essential for a student's productivity. The
comfortable keyboard and bright screen further enhance the usability of the laptop. If you ever encounter any issues or have que
stions about the laptop, please feel free to reach out for support. We're here to ensure that your experience continues to be po
sitive. Thank you for choosing our product and taking the time to share your satisfaction!


### Response:
The customer purchased a laptop for their son who is studying engineering. He is happy with its battery life, performance, sleek
design, and comfortable keyboard. The laptop comes with a warranty and free antivirus software. The customer thinks it's a great
value for money. We're glad that the customer's satisfaction with our product is evident, and we're glad to ensure that their ex
perience continues to be positive. Thank you for your purchase and for your enthusiastic review!


###

### Instruction:
Summarize the following dialogue

### Input:
```

In [41]:
```
test_summary
```

Out[41]:
```
'The user purchased a laptop for their son, who is studying engineering. They are satisfied with its battery life, fast performa
nce, sleek design, comfortable keyboard, and bright screen, and its one-year warranty.'
```

The following code allows us to run shell commands on Colab (we need shell commands to check file sizes of the estimated models).

In [42]:
```
import locale
```

In [43]:
```
def getpreferredencoding(do_setlocale = True):
    return "UTF-8"
```

```
locale.getpreferredencoding = getpreferredencoding
```

In [44]:
```
lora_model_name = "dialogue-summarizer-llama"
```

2.4 (B) Save Model

In [45]:
```
model.save_pretrained(lora_model_name)
```

In [46]:
```
!ls -lh {lora_model_name}
```

```
total 153M
-rw-r--r-- 1 root root  727 Sep 25 06:13 adapter_config.json
-rw-r--r-- 1 root root 153M Sep 25 06:13 adapter_model.safetensors
-rw-r--r-- 1 root root 5.0K Sep 25 06:13 README.md
```

In [47]:
```
!cp -r {lora_model_name} /data/
```

2.5 Evaluate Model Performance (10 Marks)

(A) Load Base Model (2 Marks)

(B) Evaluate Performance of Base Model (3 Marks)

(C) Load Trained Model (2 Marks)

(D) Evaluate Performance of Trained Model (3 Marks)

Llama 2 Base Model Performance

2.5 (A) Load Base Model

In [48]:
```
torch.cuda.empty_cache()
```

In [ ]:
```

```

In [49]:
```
baseline_model, tokenizer = FastLanguageModel.from_pretrained(
    model_name=lora_model_name,
```

```
    max_seq_length=2048,
    dtype=None, # None for auto detection. Float16 for Tesla T4, V100, Bfloat16 for Ampere+
    load_in_4bit=True # Use 4bit quantization to reduce memory usage.
)
```

```
==((====))==  Unsloth 2024.9.post2: Fast Llama patching. Transformers = 4.44.2.
   \\   /|    GPU: Tesla T4. Max memory: 14.748 GB. Platform = Linux.
O^O/ \_/ \    Pytorch: 2.4.0+cu121. CUDA = 7.5. CUDA Toolkit = 12.1.
\        /    Bfloat16 = FALSE. FA [Xformers = 0.0.28.post1. FA2 = False]
 "-____-"     Free Apache license: http://github.com/unslothai/unsloth
Unsloth: Fast downloading is enabled - ignore downloading bars which are red colored!
```

In [50]:
```
FastLanguageModel.for_inference(baseline_model)
```

Out[50]:

```
PeftModelForCausalLM(
  (base_model): LoraModel(
    (model): LlamaForCausalLM(
      (model): LlamaModel(
        (embed_tokens): Embedding(32000, 4096)
        (layers): ModuleList(
          (0-31): 32 x LlamaDecoderLayer(
            (self_attn): LlamaAttention(
              (q_proj): lora.Linear4bit(
                (base_layer): Linear4bit(in_features=4096, out_features=4096, bias=False)
                (lora_dropout): ModuleDict(
                  (default): Identity()
                )
                (lora_A): ModuleDict(
                  (default): Linear(in_features=4096, out_features=16, bias=False)
                )
                (lora_B): ModuleDict(
                  (default): Linear(in_features=16, out_features=4096, bias=False)
                )
                (lora_embedding_A): ParameterDict()
                (lora_embedding_B): ParameterDict()
                (lora_magnitude_vector): ModuleDict()
              )
              (k_proj): lora.Linear4bit(
                (base_layer): Linear4bit(in_features=4096, out_features=4096, bias=False)
                (lora_dropout): ModuleDict(
                  (default): Identity()
                )
                (lora_A): ModuleDict(
                  (default): Linear(in_features=4096, out_features=16, bias=False)
                )
                (lora_B): ModuleDict(
                  (default): Linear(in_features=16, out_features=4096, bias=False)
                )
                (lora_embedding_A): ParameterDict()
                (lora_embedding_B): ParameterDict()
                (lora_magnitude_vector): ModuleDict()
              )
              (v_proj): lora.Linear4bit(
                (base_layer): Linear4bit(in_features=4096, out_features=4096, bias=False)
                (lora_dropout): ModuleDict(
                  (default): Identity()
                )
                (lora_A): ModuleDict(
```

```
      (default): Linear(in_features=4096, out_features=16, bias=False)
    )
    (lora_B): ModuleDict(
      (default): Linear(in_features=16, out_features=4096, bias=False)
    )
    (lora_embedding_A): ParameterDict()
    (lora_embedding_B): ParameterDict()
    (lora_magnitude_vector): ModuleDict()
  )
  (o_proj): lora.Linear4bit(
    (base_layer): Linear4bit(in_features=4096, out_features=4096, bias=False)
    (lora_dropout): ModuleDict(
      (default): Identity()
    )
    (lora_A): ModuleDict(
      (default): Linear(in_features=4096, out_features=16, bias=False)
    )
    (lora_B): ModuleDict(
      (default): Linear(in_features=16, out_features=4096, bias=False)
    )
    (lora_embedding_A): ParameterDict()
    (lora_embedding_B): ParameterDict()
    (lora_magnitude_vector): ModuleDict()
  )
  (rotary_emb): LlamaRotaryEmbedding()
)
(mlp): LlamaMLP(
  (gate_proj): lora.Linear4bit(
    (base_layer): Linear4bit(in_features=4096, out_features=11008, bias=False)
    (lora_dropout): ModuleDict(
      (default): Identity()
    )
    (lora_A): ModuleDict(
      (default): Linear(in_features=4096, out_features=16, bias=False)
    )
    (lora_B): ModuleDict(
      (default): Linear(in_features=16, out_features=11008, bias=False)
    )
    (lora_embedding_A): ParameterDict()
    (lora_embedding_B): ParameterDict()
    (lora_magnitude_vector): ModuleDict()
  )
  (up_proj): lora.Linear4bit(
    (base_layer): Linear4bit(in_features=4096, out_features=11008, bias=False)
```

```
                (lora_dropout): ModuleDict(
                  (default): Identity()
                )
                (lora_A): ModuleDict(
                  (default): Linear(in_features=4096, out_features=16, bias=False)
                )
                (lora_B): ModuleDict(
                  (default): Linear(in_features=16, out_features=11008, bias=False)
                )
                (lora_embedding_A): ParameterDict()
                (lora_embedding_B): ParameterDict()
                (lora_magnitude_vector): ModuleDict()
              )
              (down_proj): lora.Linear4bit(
                (base_layer): Linear4bit(in_features=11008, out_features=4096, bias=False)
                (lora_dropout): ModuleDict(
                  (default): Identity()
                )
                (lora_A): ModuleDict(
                  (default): Linear(in_features=11008, out_features=16, bias=False)
                )
                (lora_B): ModuleDict(
                  (default): Linear(in_features=16, out_features=4096, bias=False)
                )
                (lora_embedding_A): ParameterDict()
                (lora_embedding_B): ParameterDict()
                (lora_magnitude_vector): ModuleDict()
              )
              (act_fn): SiLU()
            )
            (input_layernorm): LlamaRMSNorm((4096,), eps=1e-05)
            (post_attention_layernorm): LlamaRMSNorm((4096,), eps=1e-05)
          )
        )
        (norm): LlamaRMSNorm((4096,), eps=1e-05)
        (rotary_emb): LlamaRotaryEmbedding()
      )
      (lm_head): Linear(in_features=4096, out_features=32000, bias=False)
    )
  )
)
```

In [51]:
```
test_dataset = datasets.Dataset.from_pandas(sample_reviews_gold_examples_df)
```

Single Inference

In [52]:
```
torch.cuda.empty_cache()
```

In [53]:
```
instruction = "Summarize the following dialogue"
test_dialogue = test_dataset[0]['dialogue']
test_summary = test_dataset[0]['summary']
```

In [54]:
```
input = tokenizer(
    alpaca_prompt.format(
        instruction,
        test_dialogue,
        ""
    ), return_tensors="pt"
).to("cuda")
```

In [55]:
```
output = baseline_model.generate(
    **input,
    max_new_tokens=128,
    use_cache=True,
    do_sample=True,
    temperature=0.2
)
```

In [56]:
```
print(tokenizer.decode(output[0]))
```

```
<s> Below is an instruction that describes a task, paired with an input that provides further context. Write a response that app
ropriately completes the request.

### Instruction:
Summarize the following dialogue

### Input:
customer : I bought this laptop for my son who is studying engineering. He is very happy with it. It has a good battery life, fa
st performance, and a sleek design. The keyboard is comfortable and the screen is bright. The laptop came with a one-year warran
ty and a free antivirus software. I think it is a great value for money.
response : It's fantastic to hear that the laptop you purchased for your son has met his needs and expectations, especially in h
is engineering studies. A good battery life, fast performance, and sleek design are essential for a student's productivity. The
comfortable keyboard and bright screen further enhance the usability of the laptop. If you ever encounter any issues or have que
stions about the laptop, please feel free to reach out for support. We're here to ensure that your experience continues to be po
sitive. Thank you for choosing our product and taking the time to share your satisfaction!


### Response:
The customer purchased a laptop for their son who is studying engineering. He is happy with its battery life, performance, sleek
design, and comfortable keyboard. The laptop comes with a warranty and free antivirus software. The customer thinks it's a great
value for money. We're glad that the laptop meets the customer's needs and expectations. If they have any issues, we're here to
support. We're glad the customer chose our product and thank them for their satisfaction response


###

### Instruction:
Summarize the following dialogue

###
```

Batch Inference

In [57]:
```
torch.cuda.empty_cache()
```

In [58]:
```
instruction = "Summarize the following dialogue"
```

In [59]:
```
test_dialogues = [sample['dialogue'] for sample in test_dataset]
test_summaries = [sample['summary'] for sample in test_dataset]
```

In [60]:
```
def extract_summary_from_string(input_string):
    try:
```

```
        # Assuming the response is between ### Response: and </s>
        summary_start = input_string.rfind('### Response:\n') + 14 # number of characters in '### Response:\n'
        summary_end = input_string.rfind('</s>')
        summary_str = input_string[summary_start:summary_end]

        return summary_str
    except Exception as e:
        print(f"Error decoding string: {e}")
        return None
```

In [61]:
```
predicted_summaries = []
```

In [62]:
```
for sample_dialogue in test_dialogues:
    input = tokenizer(
    alpaca_prompt.format(
        instruction,
        sample_dialogue,
        ""
    ), return_tensors="pt"
  ).to("cuda")

    outputs = baseline_model.generate(
    **input,
    max_new_tokens=256,
    use_cache=True
  )

    predicted_summary = tokenizer.decode(outputs[0])

    output_str = extract_summary_from_string(predicted_summary)

    predicted_summaries.append(output_str)
```

Evaluate

2.5 (B) Evaluate Performance of Base Model

In [63]:
```
bert_scorer = evaluate.load("bertscore")
```

```
Downloading builder script:   0%|          | 0.00/7.95k [00:00<?, ?B/s]
```

In [64]:
```
bert_scorer
```

Out[64]:
```
EvaluationModule(name: "bert_score", module_type: "metric", features: [{'predictions': Value(dtype='string', id='sequence'), 're
ferences': Sequence(feature=Value(dtype='string', id='sequence'), length=-1, id='references')}, {'predictions': Value(dtype='str
ing', id='sequence'), 'references': Value(dtype='string', id='sequence')}], usage: """
BERTScore Metrics with the hashcode from a source against one or more references.

Args:
    predictions (list of str): Prediction/candidate sentences.
    references (list of str or list of list of str): Reference sentences.
    lang (str): Language of the sentences; required (e.g. 'en').
    model_type (str): Bert specification, default using the suggested
        model for the target language; has to specify at least one of
        `model_type` or `lang`.
    num_layers (int): The layer of representation to use,
        default using the number of layers tuned on WMT16 correlation data.
    verbose (bool): Turn on intermediate status update.
    idf (bool or dict): Use idf weighting; can also be a precomputed idf_dict.
    device (str): On which the contextual embedding model will be allocated on.
        If this argument is None, the model lives on cuda:0 if cuda is available.
    nthreads (int): Number of threads.
    batch_size (int): Bert score processing batch size,
        at least one of `model_type` or `lang`. `lang` needs to be
        specified when `rescale_with_baseline` is True.
    rescale_with_baseline (bool): Rescale bertscore with pre-computed baseline.
    baseline_path (str): Customized baseline file.
    use_fast_tokenizer (bool): `use_fast` parameter passed to HF tokenizer. New in version 0.3.10.

Returns:
    precision: Precision.
    recall: Recall.
    f1: F1 score.
    hashcode: Hashcode of the library.

Examples:

    >>> predictions = ["hello there", "general kenobi"]
    >>> references = ["hello there", "general kenobi"]
    >>> bertscore = evaluate.load("bertscore")
    >>> results = bertscore.compute(predictions=predictions, references=references, lang="en")
    >>> print([round(v, 2) for v in results["f1"]])
    [1.0, 1.0]
""", stored examples: 0)
```

In [65]:
```
# Provide Prediction Summaries and Test Summaries as input
score = bert_scorer.compute(
    predictions=predicted_summaries,
    references= test_summaries,
    lang="en",
    rescale_with_baseline=True
)
```

```
tokenizer_config.json:   0%|          | 0.00/25.0 [00:00<?, ?B/s]
config.json:   0%|          | 0.00/482 [00:00<?, ?B/s]
vocab.json:   0%|          | 0.00/899k [00:00<?, ?B/s]
merges.txt:   0%|          | 0.00/456k [00:00<?, ?B/s]
tokenizer.json:   0%|          | 0.00/1.36M [00:00<?, ?B/s]
```

```
/usr/local/lib/python3.10/dist-packages/transformers/tokenization_utils_base.py:1601: FutureWarning: `clean_up_tokenization_spaces` was not set. It will be set to `True` by default. This behavior will be depracted in transformers v4.45, and will be then set to `False` by default. For more details check this issue: https://github.com/huggingface/transformers/issues/31884
  warnings.warn(
```

```
model.safetensors:   0%|          | 0.00/1.42G [00:00<?, ?B/s]
```

```
Some weights of RobertaModel were not initialized from the model checkpoint at roberta-large and are newly initialized: ['roberta.pooler.dense.bias', 'roberta.pooler.dense.weight']
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
```

In [66]:
```
#  Calculate Average Bert Score . Average Bert Score is sum of f1 score divided by number of samples
f1_scores = score["f1"]  # This should be a list of F1 scores

# Calculate the average BERTScore (F1 score)
average_bert_score = sum(f1_scores) / len(f1_scores) if f1_scores else 0

print("Average BERTScore (F1):", average_bert_score)
```

```
Average BERTScore (F1): 0.4318865239620209
```

Llama 2 Trained (Fine-tuned) Model Summarizer

In [67]:
```
lora_model_name = "dialogue-summarizer-llama"
```

In [68]:
```
def getpreferredencoding(do_setlocale = True):
    return "UTF-8"

locale.getpreferredencoding = getpreferredencoding
```

In [71]:
```
!cp -r {lora_model_name} /data/
```

In [72]:
```
#!cp -r /data/{lora_model_name} .
```

2.5 (C) Load Trained Model

In [73]:
```
model, tokenizer = FastLanguageModel.from_pretrained(
    model_name=lora_model_name,
    max_seq_length=2048,
    dtype=None,
    load_in_4bit=True
)
```

```
==((====))==  Unsloth 2024.9.post2: Fast Llama patching. Transformers = 4.44.2.
   \\   /|    GPU: Tesla T4. Max memory: 14.748 GB. Platform = Linux.
O^O/ \_/ \    Pytorch: 2.4.0+cu121. CUDA = 7.5. CUDA Toolkit = 12.1.
\        /    Bfloat16 = FALSE. FA [Xformers = 0.0.28.post1. FA2 = False]
 "-____-"     Free Apache license: http://github.com/unslothai/unsloth
Unsloth: Fast downloading is enabled - ignore downloading bars which are red colored!
```

In [74]:
```
FastLanguageModel.for_inference(model)
```

Out[74]:

```
PeftModelForCausalLM(
  (base_model): LoraModel(
    (model): LlamaForCausalLM(
      (model): LlamaModel(
        (embed_tokens): Embedding(32000, 4096)
        (layers): ModuleList(
          (0-31): 32 x LlamaDecoderLayer(
            (self_attn): LlamaAttention(
              (q_proj): lora.Linear4bit(
                (base_layer): Linear4bit(in_features=4096, out_features=4096, bias=False)
                (lora_dropout): ModuleDict(
                  (default): Identity()
                )
                (lora_A): ModuleDict(
                  (default): Linear(in_features=4096, out_features=16, bias=False)
                )
                (lora_B): ModuleDict(
                  (default): Linear(in_features=16, out_features=4096, bias=False)
                )
                (lora_embedding_A): ParameterDict()
                (lora_embedding_B): ParameterDict()
                (lora_magnitude_vector): ModuleDict()
              )
              (k_proj): lora.Linear4bit(
                (base_layer): Linear4bit(in_features=4096, out_features=4096, bias=False)
                (lora_dropout): ModuleDict(
                  (default): Identity()
                )
                (lora_A): ModuleDict(
                  (default): Linear(in_features=4096, out_features=16, bias=False)
                )
                (lora_B): ModuleDict(
                  (default): Linear(in_features=16, out_features=4096, bias=False)
                )
                (lora_embedding_A): ParameterDict()
                (lora_embedding_B): ParameterDict()
                (lora_magnitude_vector): ModuleDict()
              )
              (v_proj): lora.Linear4bit(
                (base_layer): Linear4bit(in_features=4096, out_features=4096, bias=False)
                (lora_dropout): ModuleDict(
                  (default): Identity()
                )
                (lora_A): ModuleDict(
```

```
      (default): Linear(in_features=4096, out_features=16, bias=False)
    )
    (lora_B): ModuleDict(
      (default): Linear(in_features=16, out_features=4096, bias=False)
    )
    (lora_embedding_A): ParameterDict()
    (lora_embedding_B): ParameterDict()
    (lora_magnitude_vector): ModuleDict()
  )
  (o_proj): lora.Linear4bit(
    (base_layer): Linear4bit(in_features=4096, out_features=4096, bias=False)
    (lora_dropout): ModuleDict(
      (default): Identity()
    )
    (lora_A): ModuleDict(
      (default): Linear(in_features=4096, out_features=16, bias=False)
    )
    (lora_B): ModuleDict(
      (default): Linear(in_features=16, out_features=4096, bias=False)
    )
    (lora_embedding_A): ParameterDict()
    (lora_embedding_B): ParameterDict()
    (lora_magnitude_vector): ModuleDict()
  )
  (rotary_emb): LlamaRotaryEmbedding()
)
(mlp): LlamaMLP(
  (gate_proj): lora.Linear4bit(
    (base_layer): Linear4bit(in_features=4096, out_features=11008, bias=False)
    (lora_dropout): ModuleDict(
      (default): Identity()
    )
    (lora_A): ModuleDict(
      (default): Linear(in_features=4096, out_features=16, bias=False)
    )
    (lora_B): ModuleDict(
      (default): Linear(in_features=16, out_features=11008, bias=False)
    )
    (lora_embedding_A): ParameterDict()
    (lora_embedding_B): ParameterDict()
    (lora_magnitude_vector): ModuleDict()
  )
  (up_proj): lora.Linear4bit(
    (base_layer): Linear4bit(in_features=4096, out_features=11008, bias=False)
```

```
                  (lora_dropout): ModuleDict(
                    (default): Identity()
                  )
                  (lora_A): ModuleDict(
                    (default): Linear(in_features=4096, out_features=16, bias=False)
                  )
                  (lora_B): ModuleDict(
                    (default): Linear(in_features=16, out_features=11008, bias=False)
                  )
                  (lora_embedding_A): ParameterDict()
                  (lora_embedding_B): ParameterDict()
                  (lora_magnitude_vector): ModuleDict()
                )
                (down_proj): lora.Linear4bit(
                  (base_layer): Linear4bit(in_features=11008, out_features=4096, bias=False)
                  (lora_dropout): ModuleDict(
                    (default): Identity()
                  )
                  (lora_A): ModuleDict(
                    (default): Linear(in_features=11008, out_features=16, bias=False)
                  )
                  (lora_B): ModuleDict(
                    (default): Linear(in_features=16, out_features=4096, bias=False)
                  )
                  (lora_embedding_A): ParameterDict()
                  (lora_embedding_B): ParameterDict()
                  (lora_magnitude_vector): ModuleDict()
                )
                (act_fn): SiLU()
              )
              (input_layernorm): LlamaRMSNorm((4096,), eps=1e-05)
              (post_attention_layernorm): LlamaRMSNorm((4096,), eps=1e-05)
            )
          )
          (norm): LlamaRMSNorm((4096,), eps=1e-05)
          (rotary_emb): LlamaRotaryEmbedding()
        )
        (lm_head): Linear(in_features=4096, out_features=32000, bias=False)
      )
    )
  )
```

In [75]:
```
test_dataset = datasets.Dataset.from_pandas(sample_reviews_gold_examples_df)
```

Single Inference

In [76]:
```
torch.cuda.empty_cache()
```

In [77]:
```
instruction = "Summarize the following dialogue"
test_dialogue = test_dataset[0]['dialogue']
test_summary = test_dataset[0]['summary']
```

In [78]:
```
input = tokenizer(
    alpaca_prompt.format(
        instruction,
        test_dialogue,
        ""
    ), return_tensors="pt"
).to("cuda")
```

In [79]:
```
output = model.generate(
    **input,
    max_new_tokens=128,
    use_cache=True,
    do_sample=True,
    temperature=0.2
)
```

In [80]:
```
print(tokenizer.decode(output[0]))
```

```
<s> Below is an instruction that describes a task, paired with an input that provides further context. Write a response that app
ropriately completes the request.

### Instruction:
Summarize the following dialogue

### Input:
customer : I bought this laptop for my son who is studying engineering. He is very happy with it. It has a good battery life, fa
st performance, and a sleek design. The keyboard is comfortable and the screen is bright. The laptop came with a one-year warran
ty and a free antivirus software. I think it is a great value for money.
response : It's fantastic to hear that the laptop you purchased for your son has met his needs and expectations, especially in h
is engineering studies. A good battery life, fast performance, and sleek design are essential for a student's productivity. The
comfortable keyboard and bright screen further enhance the usability of the laptop. If you ever encounter any issues or have que
stions about the laptop, please feel free to reach out for support. We're here to ensure that your experience continues to be po
sitive. Thank you for choosing our product and taking the time to share your satisfaction!


### Response:
The customer purchased a laptop for their son who is studying engineering. He is happy with its battery life, performance, sleek
design, and comfortable keyboard. The laptop comes with a warranty and free antivirus software. The customer thinks it's a great
value for money. We're glad that the customer's satisfaction with our product is evident, and we're glad to ensure that their ex
perience continues to be positive. Thank you for your purchase and for your enthusiastic review!


###

### Instruction:
Summarize the following dialogue

### Input:
```

Batch Inference

In [81]:
```
torch.cuda.empty_cache()
```

In [82]:
```
instruction = "Summarize the following dialogue"
```

In [ ]:
```
# test_size = 4
```

In [83]:
```
test_dialogues = [sample['dialogue'] for sample in test_dataset]
test_summaries = [sample['summary'] for sample in test_dataset]
```

In [101…
```
len(test_dialogues)
```

Out[101]:
```
4
```

In [100…
```
len(test_summaries)
```

Out[100]:
```
4
```

In [102…
```
def extract_summary_from_string(input_string):
    try:
        # Assuming the response is between ### Response: and </s>
        summary_start = input_string.rfind('### Response:\n') # number of characters in '### Response:\n'
        summary_end = input_string.rfind('</s>')
        summary_str = input_string[summary_start:summary_end]

        return summary_str
    except Exception as e:
        print(f"Error decoding string: {e}")
        return None
```

In [103…
```
predicted_summaries = []
```

In [104…
```
for sample_dialogue in test_dialogues:
    input = tokenizer(
    alpaca_prompt.format(
        instruction,
        sample_dialogue,
        ""
    ), return_tensors="pt"
  ).to("cuda")

    outputs = baseline_model.generate(
    **input,
    max_new_tokens=256,
    use_cache=True
  )

    predicted_summary = tokenizer.decode(outputs[0])

    output_str = extract_summary_from_string(predicted_summary)
```

```
    predicted_summaries.append(output_str)
```

In [107…

In [105…
```
len(predicted_summaries)
```

Out[105]:
```
4
```

Evaluate

2.5 (D) Evaluate Performance of Trained Model

In [106…
```
# Input prediction summaries and test summaries in bert scorer
score = bert_scorer.compute(
    predictions=predicted_summaries,
    references=test_summaries,
    lang="en",
    rescale_with_baseline=True
)
```

In [107…
```
#  Calculate Average Bert Score . Average Bert Score is sum of f1 score divided by number of samples
#  Calculate Average Bert Score . Average Bert Score is sum of f1 score divided by number of samples
f1_scores = score["f1"]  # This should be a list of F1 scores

# Calculate the average BERTScore (F1 score)
average_bert_score = sum(f1_scores) / len(f1_scores) if f1_scores else 0

print("Average BERTScore (F1):", average_bert_score)
```

```
Average BERTScore (F1): 0.28979703411459923
```

Summary

BERT Score:

- Baseline (BERT): 43.18%
- Trained Model: 29%

Insights

1. Performance Gap:

- There is a notable performance gap of 14.18% between the BERT score and the trained model score. This indicates that the trained model is significantly underperforming compared to the baseline. Potential Reasons for Underperformance:

1. Training Data Quality: The quality and relevance of the training data can greatly affect model performance. If the training data does not cover a wide range of examples or is noisy, it can lead to poor generalization. Model Configuration: The hyperparameters used during training (such as learning rate, batch size, and training steps) may not be optimal for your specific task, leading to inadequate learning.

2. Overfitting: If the model was trained for too long or on a dataset that does not represent the real-world application, it may have memorized the training data instead of learning to generalize. Complexity of Task: The task's complexity might require a more sophisticated model architecture or additional training technique

Next Steps:

- **Reassess Training Data:** Review the training dataset to ensure it is clean, diverse, and representative of the tasks the model will perform in production. Consider augmenting the dataset if necessary.

-**Hyperparameter Tuning:** Experiment with different hyperparameters, such as learning rate, batch size, and the number of training epochs, to find the most effective combination.

-**Model Architecture:** If the current model architecture does not yield satisfactory results, consider exploring other architectures or incorporating more advanced techniques, like transfer learning or ensemble methods.

Evaluation Metrics:

It may also be beneficial to look beyond the BERT score and evaluate the model using other metrics like accuracy, F1 score, or BLEU score, depending on the specific task, to gain a more comprehensive understanding of its performance.

Conclusion

The significant difference between the BERT score and the trained model score highlights the need for further investigation and adjustment of your training process. By reassessing your data quality, tuning hyperparameters, and potentially modifying the model architecture, you can work towards improving your trained model's performance.

==========END ==========

In [ ]:

In [ ]: