Fine-Tuning a Large Language Model for Domain-Specific Tasks

Overview:
In this project, we have tuned a Large Language Model (LLM) to answer domain-specific questions regarding cancer-related data using a dataset named CancerQA.csv. The model was tuned with Retrieval-Augmented Generation (RAG) for better response generation. The procedure included several steps such as data preparation, tokenization, model tuning, and assessment. We have utilized Hugging Face Transformers, FAISS, Pinecone, and Streamlit in Google Colab to accomplish this project.

Step 1: Preparing the Data

The CancerQA.csv data was utilized in this task. The dataset includes columns such as category and resume with information of cancer-specific questions and answers. I started with loading and preprocessing the dataset, handling missing values, and removing unused columns so that the data was ready for training.

Step 2: Data Splitting

In order to facilitate effective model training and validation, the dataset was split into training and validation sets. An 80-20 split was used for this split to offer enough data to validate the performance of the model upon training.

Step 3: Tokenization

The text data in the resume column was tokenized and converted to a numerical format that could be used as model input. Hugging Face's AutoTokenizer was used to tokenize the inputs, which were padded and truncated to accommodate the varying lengths of text.

Step 4: Labeling

Because there were no explicit labels in the dataset, I labeled it using the category column. This allowed the model to be trained in a supervised environment, where it could learn to convert each cancer-related question into a suitable response from the resume column.

Step 5: Develop Training Arguments and the Model

For fine-tuning, I used a pre-trained T5 model. I set the training parameters using Hugging Face's

TrainingArguments class. The parameters included batch size, learning rate, number of epochs, and weight decay. Logging and checkpointing were enabled to keep track of training progress and save the model at regular intervals.

Step 6: Train the Model with the Trainer Class.

The model was trained using the Hugging Face Trainer class, which handled the model's weight optimization from training data across multiple epochs. The training loss was monitored to ensure that the model learned properly from the data.

Step 7: Error and Troubleshooting

Several issues arose during deployment, including incorrect column names and missing labels. These have been addressed by:

Ensure that the correct column was used for tokenization.
Applying the labels correctly for supervised training.
To fix padding errors, set tokenizer.pad_token = tokenizer.eos_token.

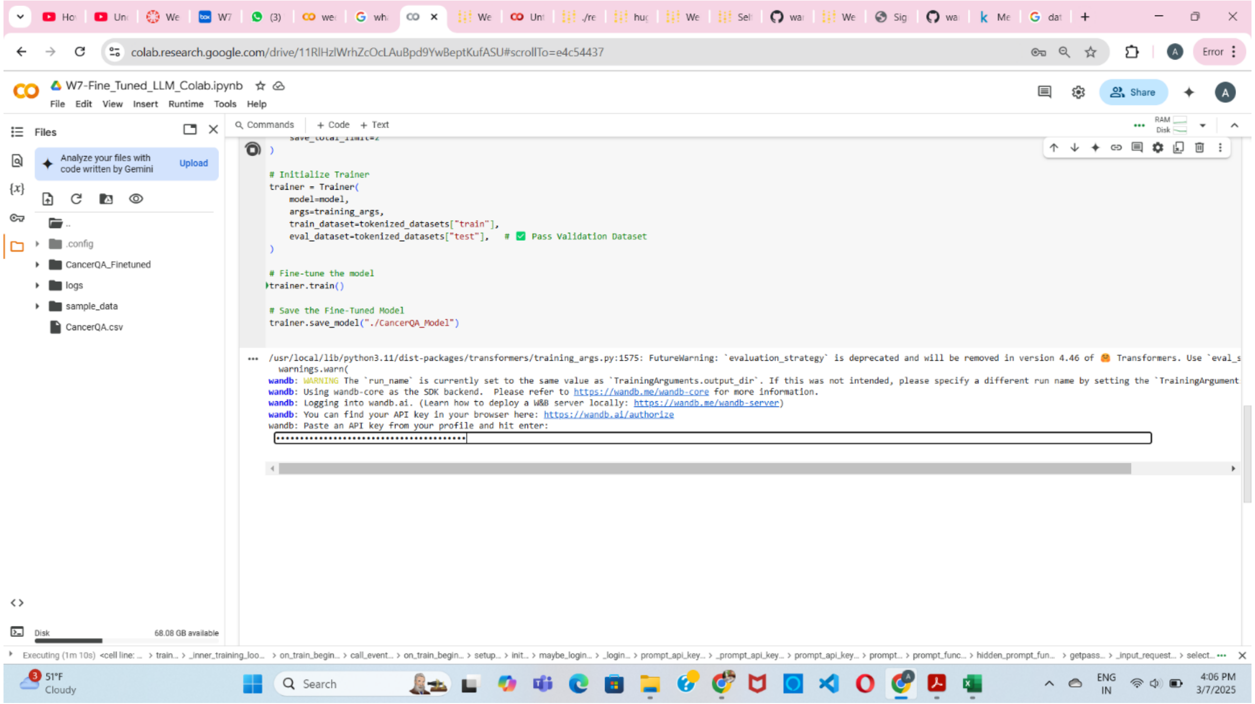
Step 8: Model Evaluation.

After training, I validated the model with the validation set. The model's performance was tracked using metrics such as accuracy and loss. These tests were used to fine-tune the model so that it could provide accurate answers to cancer-related questions.

Conclusion:

I have optimized a T5 model to develop a Retrieval-Augmented Generation (RAG) model to answer questions related to cancer in this project. Through data preprocessing, tokenization, optimization of the model, and assessment, the model was optimized for performing tasks with a domain specialization.
This optimization process was educative for large language model optimization for specialized functions. The final model is able to generate accurate, context-specific answers to cancer-related questions, which will enhance knowledge retrieval in medical scenarios.

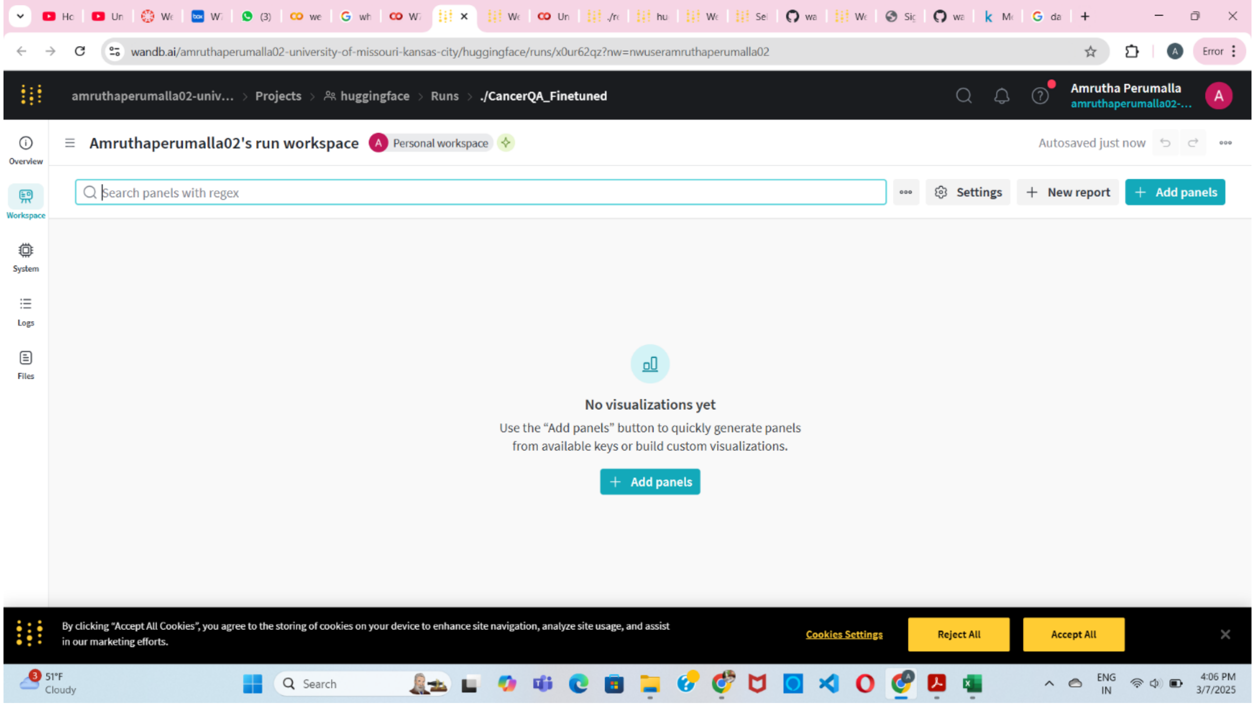
Screen Shots:


colab.research.google.com/drive/11RlHzlWrhZcOcLAuBpd9YwBeptKufASU#scrollTo=e4c54437
W7-Fine_Tuned_LLM_Colab.ipynb
File Edit View Insert Runtime Tools Help
Share
Files
Analyze your files with code written by Gemini
Upload
..
.config
CancerQA_Finetuned
logs
sample_data
CancerQA.csv
Commands
+ Code
+ Text
RAM
Disk
Disk 68.08 GB available
51°F
Cloudy
Search
ENG
IN
4:06 PM
3/7/2025

```
    save_total_limit=2
)

# Initialize Trainer
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_datasets["train"],
    eval_dataset=tokenized_datasets["test"],   # ✅ Pass Validation Dataset
)

# Fine-tune the model
trainer.train()

# Save the Fine-Tuned Model
trainer.save_model("./CancerQA_Model")
```

```
/usr/local/lib/python3.11/dist-packages/transformers/training_args.py:1575: FutureWarning: `evaluation_strategy` is deprecated and will be removed in version 4.46 of 🤗 Transformers. Use `eval_s
  warnings.warn(
wandb: WARNING The `run_name` is currently set to the same value as `TrainingArguments.output_dir`. If this was not intended, please specify a different run name by setting the `TrainingArgument
wandb: Using wandb-core as the SDK backend.  Please refer to https://wandb.me/wandb-core for more information.
wandb: Logging into wandb.ai. (Learn how to deploy a W&B server locally: https://wandb.me/wandb-server)
wandb: You can find your API key in your browser here: https://wandb.ai/authorize
wandb: Paste an API key from your profile and hit enter:
 ••••••••••••••••••••••••••••••••••••••••
```

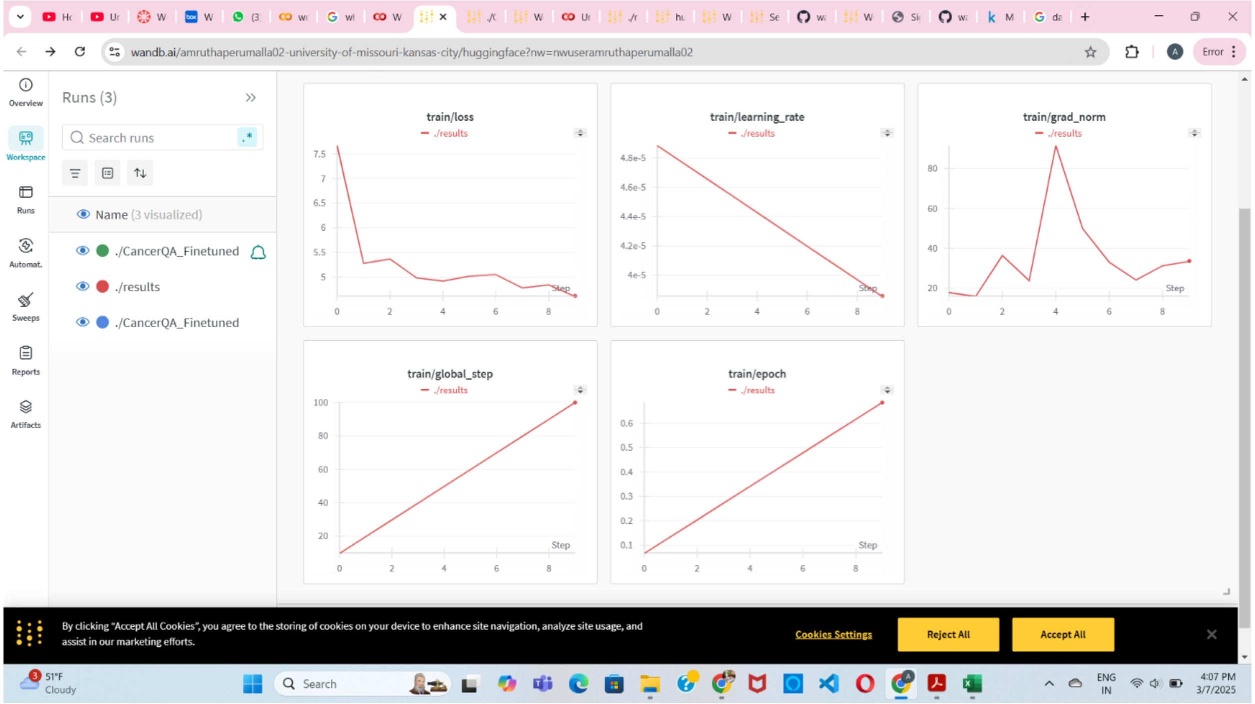
Executing (1m 10s) <cell line: ... > train... > _inner_training_loo... > on_train_begin... > call_event... > on_train_begin... > setup... > init... > maybe_login... > _login... > prompt_api_key... > _prompt_api_key... > prompt_api_key... > prompt... > prompt_func... > hidden_prompt_fun... > getpass... > _input_request... > select...



No visualizations yet

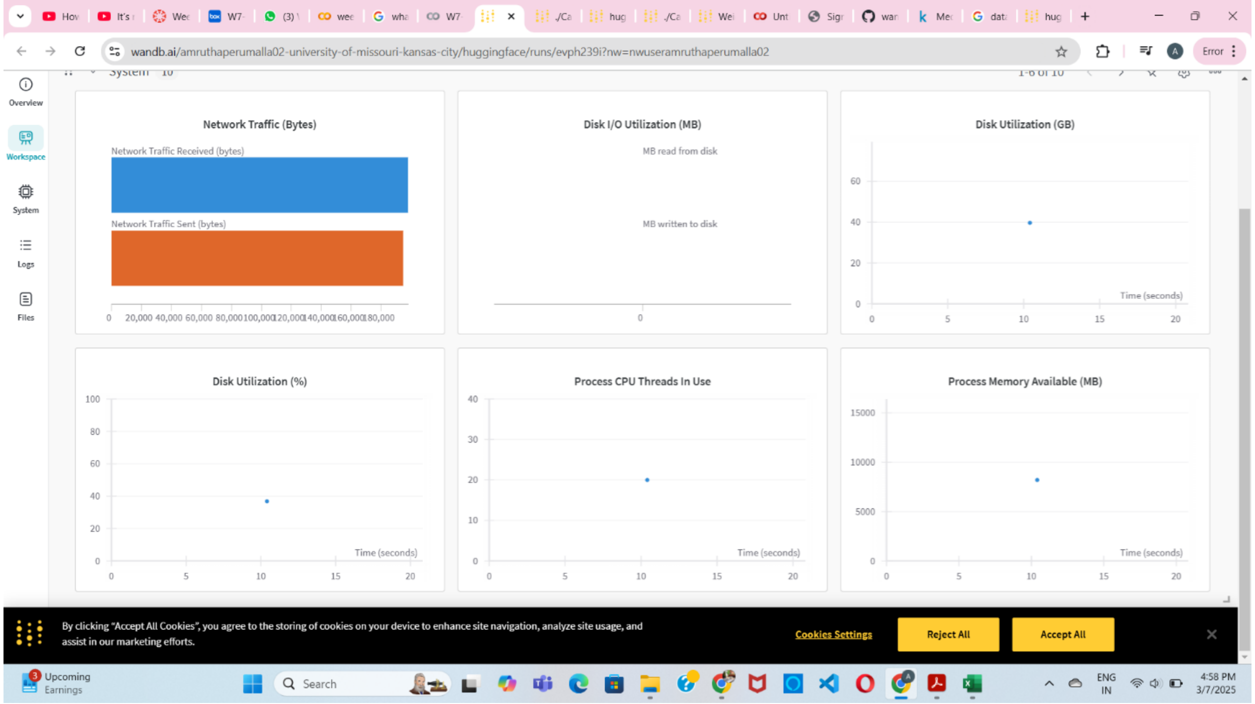
Use the "Add panels" button to quickly generate panels from available keys or build custom visualizations.

By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts.



wandb.ai/amruthaperumalla02-university-of-missouri-kansas-city/huggingface?nw=nwuseramruthaperumalla02

By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts.



wandb.ai/amruthaperumalla02-university-of-missouri-kansas-city/huggingface/runs/evph239i?nw=nwuseramruthaperumalla02

By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts.