REPORT

Fine-tuning Model:

1.Imports:

- The script starts by importing essential libraries for data handling, model training, and optimization.

- `torch` and `transformers` are used for deep learning and working with pre-trained models.

- `pandas` is used for data manipulation and handling.

- `numpy` is imported for numerical operations.

2.Device Configuration:

- It checks for GPU availability and sets the device to either GPU or CPU accordingly using `torch.device()`.

3.Data Loading and Preprocessing:

- The script loads the Amazon Reviews dataset from a CSV file using `pd.read_csv()`.

- Rows with missing values in the 'Text' and 'Summary' columns are dropped using `dropna()`.

- Data preprocessing involves concatenating the 'Text' and 'Summary' columns with the task designator "TL;DR".

4.Dataset Class:

- Defines a custom dataset class `ReviewDataset` inherited from `torch.utils.data.Dataset`.

- Handles tokenization, padding, and truncation of input sequences within the class methods `__init__`, `__len__`, `__getitem__`, and `pad_truncate`.

5.Tokenizer and Model Loading:

- Loads the pre-trained GPT-2 tokenizer and model using `AutoTokenizer.from_pretrained()` and `AutoModelWithLMHead.from_pretrained()`.

6.Training Parameters:

- Defines batch size, number of epochs, and optimization parameters (`batch_size`, `epochs`, `optimizer`).

7.Training Loop:

- Defines a training function `train` that iterates over epochs and batches, computes loss, and performs backpropagation.

- Model parameters are updated using the specified optimizer (`optim.AdamW`).

8.Training Execution:

- Loads the dataset, creates a DataLoader for batching, initializes the optimizer, and starts training the model.

9.Model Saving:

- After training, saves the fine-tuned model and tokenizer to the specified directory (`model_dir`).

Summary Generation and ROUGE Score Calculation Script

1.Imports:

- Imports libraries including `torch`, `pandas`, `numpy`, `transformers`, and `rouge` for summary generation and evaluation.

2.Device Configuration and Model Loading:

- Sets up the device and loads the fine-tuned GPT-2 tokenizer and model from the specified directory.

3.Model Inference Functions:

- Defines functions for generating summaries (`model_infer`) and computing ROUGE scores (`compute_rouge_scores`).

- The `topk` function is defined for sampling from the probability distribution.

4.User Input and Summary Generation:

- Implements a function (`generate_and_evaluate_summary`) to take user input for review text and actual summary.

- Generates a summary using the fine-tuned model and computes ROUGE scores between the generated and actual summaries.

Conclusion:

Both scripts together provide a complete pipeline for fine-tuning, generating, and evaluating summaries using a GPT-2 model. They cover data preprocessing, model training, inference, and evaluation stages. With proper setup and valid input data, these scripts should effectively train a model and provide summaries along with ROUGE scores for evaluation.