Inference on CPU and Fine-tuning of LLM Model to Create a Custom Chatbot

Problem Statement

This problem statement is designed to introduce beginners to the exciting field of Generative Artificial Intelligence (GenAI) through a series of hands-on exercises. Participants will learn the basics of GenAI, perform simple Large Language Model (LLM) inference on a CPU, and explore the process of fine-tuning an LLM model to create a custom Chatbots.

Unique Idea Brief (Solution)

Focusing on both inference with a pretrained model and fine-tuning for customized tasks. It begins by initializing a sequence-to-sequence transformer model (`sshleifer/distilbart-cnn-12-6`) and demonstrates how to generate summaries from input texts, showcasing the model's capability in natural language processing tasks such as summarization.

Next, the code integrates a dataset (`samsum`) specifically designed for dialogue summarization tasks. It preprocesses this dataset for fine-tuning a sequence-to-sequence model (`sshleifer/distilbart-cnn-6-6`) using the `transformers` library's `Trainer` module. This process includes configuring training parameters, handling data collation, and executing the fine-tuning process. Once trained, the fine-tuned model is capable of generating concise summaries from dialogues, addressing practical applications in chatbots or summarization tools. Additionally, the code includes a simple command-line interface allowing users to input dialogues interactively and receive instant summaries, enhancing usability for real-world applications.

Features Offered

The features offered are to enable it to be used effectively for conversation summarization in various contexts, such as summarizing meetings or any other conversational interactions:

1. **Dialogue Summarization:** The code utilizes a fine-tuned transformer model (`sshleifer/distilbart-cnn-6-6`) specifically designed for sequence-to-sequence tasks like dialogue summarization. This model is adept at processing conversational data and generating concise summaries that capture the essence of dialogues.
2. **Application in Meetings:** It can be employed to summarize meeting transcripts, extracting key points and discussions into digestible summaries. This capability helps in condensing lengthy discussions and ensuring that important information is retained without overwhelming details.
3. **Efficient Information Extraction:** The sequence-to-sequence architecture efficiently processes input dialogues, identifying significant information and generating coherent summaries. This functionality is crucial for applications requiring quick and accurate extraction of insights from conversational data.

4. **Real-time Interaction:** With a command-line interface (CLI) for interactive use, users can input meeting transcripts or any conversational data and receive immediate summaries. This real-time interaction supports dynamic usage scenarios where timely information synthesis is essential.

Processflow

- **Model Selection and Initialization:**

  Initially, the `sshleifer/distilbart-cnn-6-6` model was selected for its suitability in sequence-to-sequence tasks.

- **Dataset Preparation:**

The `samsum` dataset, derived from WhatsApp conversations, was chosen for its relevance to dialogue summarization. This dataset contains pairs of dialogues and corresponding summaries, ideal for training the summarization model.

- **Fine-Tuning Process:**

The selected model (`sshleifer/distilbart-cnn-6-6`) was fine-tuned using the `samsum` dataset. This process involved configuring the model for sequence-to-sequence learning, where dialogues were mapped to their respective summaries. Hyperparameters such as learning rate, batch size, and number of epochs were tuned to optimize model performance.

- **Training and Evaluation:**

  - The fine-tuning process included training the model on a subset of the `samsum` dataset and evaluating its performance on a validation set. The `Trainer` class from the `transformers` library facilitated this, managing training epochs, batch processing, and performance evaluation metrics.

- **Model Deployment and Inference:**

  - After fine-tuning, the model was saved and deployed for inference. Using the fine-tuned model and tokenizer, a command-line interface (CLI) was set up to interactively summarize dialogues or meeting transcripts. This interface allowed users to input text and receive summarized outputs in real-time.

- **Application in Conversational Summarization:**

  - The fine-tuned model demonstrated its utility in summarizing conversations from various contexts, including meetings. It efficiently processed conversational data, extracting key information and generating concise summaries that captured the essence of the input dialogues.

Technologies used

- **Transformers Library (Hugging Face):**
    - This library was central to the implementation, providing pre-trained models like `sshleifer/distilbart-cnn-6-6` and tools for fine-tuning them. It facilitated efficient tokenization, model initialization, and sequence-to-sequence learning for dialogue summarization tasks.
- **PyTorch:**
    - PyTorch served as the deep learning framework for model training and inference. It provided the backend support for handling tensor operations, gradient computations, and optimization algorithms during the fine-tuning process.
- **Datasets Library (Hugging Face):**
    - The `datasets` library from Hugging Face enabled easy access and preprocessing of datasets like `samsum`. It streamlined data loading, splitting, and batch processing, crucial for training and evaluating the summarization model.
- **TensorboardX:**
    - TensorboardX was used for visualizing training metrics and monitoring model performance during the fine-tuning phase. It provided insights into training loss, evaluation metrics, and learning trends over epochs.
- **Python:**
    - Python served as the primary programming language for scripting the entire workflow. It facilitated seamless integration of libraries, data handling, and the development of the command-line interface (CLI) for interactive summarization.

Team members Contribution

Our contributions to this project were equally vital and complementary. Praveen conducted extensive research to identify suitable models and datasets, laying a robust foundation with `sshleifer/distilbart-cnn-6-6` and the `samsum` dataset.

Arya optimized model inference on CPU, ensuring efficient and scalable performance across different computational environments.

Tejas led the fine-tuning efforts, employing advanced sequence-to-sequence techniques to tailor the model for specific dialogue summarization tasks, enhancing its accuracy and relevance.

Our collaborative efforts were pivotal, combining our expertise in model selection, performance optimization, and fine-tuning methodologies. Together, we demonstrated the capabilities of modern AI and NLP technologies in real-world applications, showcasing the potential for dialogue summarization in various domains. Our contributions underscore the power of teamwork in tackling complex AI challenges and driving innovation forward.

Conclusion

In conclusion, the process outlined demonstrates a comprehensive approach to leveraging advanced NLP techniques for dialogue summarization using state-of-the-art models like `sshleifer/distilbart-cnn-6-6`. Beginning with dataset selection and preprocessing using the `samsum` dataset, the workflow progressed through model fine-tuning using sequence-to-sequence (seq2seq) methods. This fine-tuned model was then deployed for practical use in summarizing dialogues and conversations, showcasing its ability to condense and highlight key information effectively.

Key technologies such as Hugging Face's Transformers library, PyTorch for deep learning operations, and supporting tools like `accelerate` and TensorboardX were instrumental in achieving efficient training, evaluation, and deployment pipelines. The Python programming environment facilitated seamless integration of these technologies, culminating in a user-friendly command-line interface for interactive summarization.

Overall, this project exemplifies the power of modern AI and NLP techniques in enhancing communication and information extraction from textual data, offering significant potential for applications in meeting summaries, customer service automation, and beyond. As AI continues to advance, integrating such technologies promises to revolutionize how we interact with and derive insights from large volumes of text data in various domains.