DISTILGPT FINETUNING DOCUMENTATION

1. Introduction

This documentation outlines the process of fine-tuning the DistilGPT model on sales conversation data using TensorFlow and the Hugging Face Transformers library. The steps include data loading, parameter setting, the fine-tuning process, challenges faced, and measures taken to improve output quality.

2. Data Loading

The sales conversation data was uploaded to Hugging Face and loaded using the `load_dataset` function from the Hugging Face datasets library. The dataset was then preprocessed and tokenized using the AutoTokenizer from the Transformers library to ensure proper formatting and truncation to a specified maximum length.

3. Parameter Setting

Key parameters for the model and training process included:

- **Checkpoint**: The pre-trained model used as the base was `distilgpt2`.
- **Max Length**: This defined the maximum length for input sequences.
- **Batch Size**: The number of samples processed together in one pass.
- **Training Steps**: Total number of steps the model would be trained for.
- **Learning Rate**: Started at 5e-5 with a Polynomial Decay schedule.
- **Epochs**: Total number of iterations over the entire dataset.

All random seeds were set for reproducibility.

4. Fine-tuning Process

The fine-tuning process involved:

- **Preparing the Dataset**: Converting the tokenized data into TensorFlow datasets and batching them for efficient training.
- **Model Configuration**: Configuring the DistilGPT model to handle the sales conversation data, adjusting settings like the max length and batch size.
- **Training Loop**: Running the training loop with specified epochs, validation checks, and saving the fine-tuned model.

5. Challenges Faced and Solutions

1. **Handling Limited Computational Resources**:
    - **Solution**: Used smaller batch sizes and gradient accumulation to manage memory usage effectively.

2. **Model Overfitting**:
    - **Solution**: Implemented early stopping and monitored validation loss to prevent overfitting by halting training when performance stopped improving.
3. **Data Imbalance**:
    - **Solution**: Applied data augmentation techniques to balance the dataset, ensuring a variety of conversation contexts to improve the model's generalization.

6. Measures to Improve Output Quality

Several strategies were employed to enhance the quality of the generated text:

- **Top-k Sampling**: Limited the selection to the top k most probable next words during text generation, improving the diversity of responses.
- **Top-p (Nucleus) Sampling**: Considered only the smallest set of words whose cumulative probability exceeded a threshold p, enhancing the coherence of the generated text.
- **No Repeat N-grams**: Prevented the model from generating repetitive sequences by ensuring that n-grams did not repeat within the generated text.

These measures were implemented to refine the model's output, making it more relevant and engaging for sales conversations.

7. Conclusion

The fine-tuning of DistilGPT on sales conversation data was successfully completed, resulting in a model capable of generating relevant and coherent sales-related responses. By addressing challenges and implementing quality improvement measures, the output was optimized for better performance in real-world applications.

Appendix: Prompts Used for Generation

A variety of prompts related to sales conversations were used to evaluate and guide the fine-tuning process, including questions on customer service prioritization, product differentiation, managing finances, creating feedback-driven cultures, negotiation advice, data-driven decision making, presentation engagement, simplifying technical terms, adapting communication styles, and improving verbal communication skills. These prompts helped in tailoring the model to generate contextually appropriate and useful responses for sales scenarios. More about them in the evaluation report.