• LoRA (Low-rank Adaptation of Large Language Models): It is a Microsoft technique to fine-tune large language models without consuming as many computational resources. It freezes parts of the model and adds new layers, reducing the amount of necessary computations. This enables faster adjustment of models to specific tasks with less memory. The result is akin to full fine-tuning but significantly faster.

• Quantization: This process reduces the precision of LLMs (Large Language Models) to make them more efficient on devices with limited resources. LoftQ combines quantization with fine-tuning LoRA to enhance performance in language tasks, making it useful for deploying powerful models on devices with fewer resources.

• Distillation: This technique involves training a smaller and faster model (known as the "student model") to emulate a larger and more accurate one (the "teacher model"). The aim is to transfer the knowledge and predictive capability of the larger model to the smaller one, thereby reducing the computational resources required for deployment.

STEPS

PASO	Nombre	Descripción
1	Data preparation	It starts by collecting task-specific data for the chatbot, which, in this case, would be a series of questions and answers and open-ended conversations with different questions and answers.
2	Choosing an LLM	A pre-trained LLM should be selected for tuning. Selection criteria may include evaluating the LLM against existing benchmarks and performance indicators to find the appropriate choice.
3	fine-tuning	The LLM is adapted to your needs using preprocessed data specific to the task. Transfer learning with RLHF is a viable strategy for fine-tuning a chatbot model. RLHF (Reinforced Learning from Human Feedback) is a fine-tuning technique that uses human feedback to guide a model towards the desired output.
4	Performing a robust assessment	Once the fine-tuning is done, you can validate the model's performance using appropriate metrics. For an LLM-based chatbot, the following combined metrics can help measure its performance: • METEOR (Metric for Evaluation of Translation with Explicit Ordering):

		It's a smarter way to check if a translated text is good. Instead of solely looking for exact word matches, it also considers synonyms and paraphrases. This is useful in chatbot conversations, where there are often many correct ways to respond to a user query. • Perplexity: Perplexity measures how well a language model predicts a sample of text. In the context of a chatbot, it can be used to evaluate how well the language model predicts the sequence of words in its responses. This can indicate linguistic coherence and fluency in the generated responses. • Diversity metrics (Distinct-N): Diversity metrics like Distinct-N (Distinct-1, Distinct-2, etc.) can be quite useful in evaluating certain aspects of a chatbot, particularly regarding the variety and richness of its responses. They count the number of individual words (unigrams) and combinations of two words (bigrams) used. More variety often signifies more engaging and less repetitive text. • Human Evaluation: Human judges assess aspects such as coherence, relevance, context comprehension, empathy, and conversational fluency, which automated metrics may not fully capture. To evaluate the quality of LLM results, evaluators can use different techniques like Likert scale (from 1 to 5) to assess relevance, fluency, or informativeness. Additionally, to evaluate the accuracy of an LLM, evaluators can employ data labeling techniques to identify incorrect statements and categorize them into specific types of errors such as factual inaccuracies, thematic deviations, nonsensical responses, etc.
5	Test deployment	Deploy the model in a test environment to detect anomalies and identify problems. This process will help you correct errors in time and avoid incidents in production.
6	Prueba de despliegue	The testing phase may also include collecting feedback from some users, domain experts and other automated systems. You can use the feedback to improve the results of the chatbot by adjusting it with a more relevant data set.
7	Deployment in production	Finally, you can integrate the model with your chatbot application. Continuous monitoring and observability is advisable to quickly resolve real-world issues.

TECHNIQUES

Fine-tuning refers to the process of adjusting and enhancing a pre-trained machine learning model to better suit a specific dataset or a particular task. Here are some common techniques used in fine-tuning:

1. Modification of top layers: When adapting a pre-trained model to a new dataset or task, often the final layers are removed and replaced with customized layers that fit the specific task.

2. Freezing layers: At times, a portion or all of the layers of the pre-trained model are frozen to prevent them from being updated during training with the new dataset. This is useful when dealing with a small dataset and wanting to avoid overfitting.

3. Full fine-tuning: In some cases, allowing all or most of the layers of the pre-trained model to adjust to the new dataset is permitted. This can be beneficial if the new dataset is large and diverse.

4. Reduced learning rate: Often, when fine-tuning a pre-trained model, a smaller learning rate is used compared to the initial training. This helps avoid drastic changes in the weights of the pre-trained model.

5. Regularization: Techniques such as weight decay or dropout can be applied to prevent overfitting during fine-tuning.

6. Data augmentation: If the dataset is limited, data augmentation techniques can be applied to increase the number of training examples. This involves applying random transformations to existing data, such as rotations, translations, zoom, among others.

7. Hyperparameter exploration: Adjusting hyperparameters, such as learning rate, batch size, model architecture, etc., is crucial to achieving good fine-tuning.

8. Transfer Learning: This technique involves leveraging pre-trained models on similar tasks to enhance performance on a specific task. By using pre-trained models as a starting point, the need for training from scratch is reduced, speeding up the learning process.

LLMs IN AERONAUTICS

1. Predictive Maintenance with ChatGPT: ChatGPT can predict potential equipment failures, such as airplane engines, using data analysis, allowing for efficient scheduling of maintenance and reducing downtime.

2. Flight Control Systems Improvement with Natural Language Processing: ChatGPT facilitates interaction between pilots and air traffic controllers through natural language interfaces, translating pilots' commands into actions for the flight control system.

3. Enhanced Safety through Real-Time Monitoring and Analysis: ChatGPT analyzes real-time data from equipment and systems, detecting potential issues and generating alerts for maintenance or recommendations based on its analysis, thereby enhancing safety.

4. Voice-Controlled Interfaces for Pilots with ChatGPT: ChatGPT interprets pilots' voice commands, providing clear and concise responses, reducing human errors, and improving safety during flights.

5. Reduction of Human Errors through Automated Decision-Making: Automated decision-making with ChatGPT analyzes data and makes maintenance or flight control decisions based on previous information, generating precise alerts and recommendations.

6. Future Possibilities for ChatGPT in the Aeronautics Industry: In the future, ChatGPT could forecast weather patterns, enhance flight autonomy, and provide real-time analysis of weather conditions to improve flight safety.

7. AviationGPT for the Aeronautics Industry: AviationGPT specializes in aviation domain, empowering users to address various natural language processing issues in this specific field. It offers precise and relevant responses, significantly enhancing performance and equipping the industry to tackle more complex challenges.

8. AeroBERT: A transformer-based language model trained on documents related to the aerospace industry. Used in Rolls-Royce to address specific natural language processing (NLP) challenges in the field.

9. Aviation Safety Analysis: Application of generative language models like ChatGPT to enhance efficiency and expedite processing of aviation safety incident reports. ChatGPT is employed to generate incident synopses from narratives and compare them with real synopses from the Aviation Safety Reporting System (ASRS) dataset.

10. Identification of Human Factors Issues: ChatGPT is used to identify human factors issues in incidents and compared with issues identified by security analysts. An accuracy of 0.61 is observed, with ChatGPT taking a cautious approach in attributing human factors issues.

11. Responsibility Assessment: The model is utilized to assess responsibility in incidents, comparing the model outputs with manual evaluations performed by security analysts.