- LoRA (Low Rank Adaptation of Large Language Models): is a technique that streamlines the fine tuning of large models by representing weight updates with smaller matrices through low-rank decomposition. This significantly reduces the number of trainable parameters while keeping the pre-trained weights frozen. LoRA allows creating lightweight and portable models for various tasks, is compatible with other parameter-efficient methods, and does not add latency during inference.

(https://huggingface.co/docs/peft/main/en/conceptual_guides/lora)

- Quantization: reduces the accuracy of LLMs to make them more efficient on resource-constrained devices. LoftQ, combines quantization with LoRA fine tuning to improve performance on language tasks, which is useful for using powerful models on devices with fewer resources.

(https://huggingface.co/papers/2310.08659)

- Distillation: is a technique where a smaller, faster model (known as a "student model") is trained to emulate a larger, more accurate one (the "master model"). The goal is to transfer the knowledge and predictive capability of the larger model to the smaller model, thus reducing the computational resources required for its deployment.

(https://www.infoq.com/news/2023/10/google-distillation/)

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.

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.

METRICS

- METEOR (Metric for Translation Evaluation with Explicit Order):

This is a smarter way of checking whether a translated text is good. Instead of just looking for exact word matches, it also takes into account synonyms and paraphrases. This is good in chatbot conversations, where there are often many correct ways to answer a user's query.

METEOR is an evaluation metric for machine translation that calculates the harmonic mean of accuracy and completeness of unigrams, giving a higher weight to completeness. It also incorporates a penalty for sentences that differ significantly in length from the reference translations.

Suppose we have a reference translation "The fast brown dog jumps over the lazy fox" and a candidate translation "The fast brown dog jumps over the lazy fox".

Unigram accuracy measures the proportion of words in the candidate translation that match the reference translation, which in this case is 6/7 ≈ 0.857.

Unigram completeness measures the proportion of words in the reference translation that match the candidate translation, which is 6/8 = 0.75.

The harmonic mean is calculated as 2 * (precision * completeness) / (precision + completeness), which results in a value of 2 * (0.857 * 0.75) / (0.857 + 0.75) ≈ 0.799

(https://plainenglish.io/community/evaluating-nlp-models-a-comprehensive-guide-to-rouge-bleu-meteor-and-bertscore-metrics-d0f1b1)

- Perplexity:

Perplexity measures how well a language model predicts a text sample. In the context of a chatbot, it can be used to assess how well the language model predicts the sequence of words in its responses. This can mean linguistic consistency and fluency in the generated responses.

This implementation of perplexity is computed with log base e, as in perplexity = e**(sum(losses) / num_tokenized_tokens) following recent convention in deep learning frameworks.

(https://huggingface.co/spaces/evaluate-metric/perplexity)

- Diversity metrics (Distinct-N):

Diversity metrics, such as Distinct-N (Distinct-1, Distinct-2, etc.), can be quite useful in evaluating certain aspects of a chatbot, especially in terms of the variety and richness of its responses. They count the number of single words (unigrams) and two-word combinations (bigrams) used. More variety usually means more interesting and less repetitive text.

Distinct-1 (Unigram Distinctness): Calculates the ratio of unique unigrams to total unigrams. Distinct-1 = (Number of unique unigrams) / (Total number of unigrams).

Distinct-2 (Bigram Distinctness): Measures the ratio of unique bigrams to total bigrams.

Distinct-2 = (Number of unique bigrams)/(Total number of bigrams).

A higher value of Distinct-N indicates greater lexical diversity in the chatbot's responses. A low score could suggest that the model tends to repeat the same words or word combinations, which could negatively affect the perceived quality of the responses.

(https://aclanthology.org/2022.acl-short.86.pdf)

- ROUGE

The ROUGE (Recall-Oriented Understudy for Gisting Evaluation) method is an evaluation metric used to assess the quality of natural language processing (NLP) tasks, such as text summarization and machine translation. ROUGE measures the N-gram overlap between the system-generated summary and the reference summary, providing information about the precision and recall of the system's output. There are several variants of ROUGE, including ROUGE-N, which quantifies the N-gram overlap, and ROUGE-L, which calculates the Longest Common Substring (LCS) between the system and reference summaries.

(https://plainenglish.io/community/evaluating-nlp-models-a-comprehensive-guide-to-rouge-bleu-meteor-and-bertscore-metrics-d0f1b1)

- BLEU

The BLEU Method (Bilingual Evaluation Understudy) is a score used to compare a proposed translation of a text with one or more reference translations. Its scale ranges from 0 to 1, where 1 means that the proposed sentence perfectly matches one of the reference sentences. Although it was initially designed to evaluate translation models, it is now used in other natural language processing (NLP) applications.

(https://plainenglish.io/community/evaluating-nlp-models-a-comprehensive-guide-to-rouge-bleu-meteor-and-bertscore-metrics-d0f1b1)

- BERTScore

BERTScore is an evaluation metric for natural language processing (NLP) tasks that uses the pre-trained BERT model to measure the similarity between two sentences. It is based on the cosine similarity between contextualized embeddings of words in candidate and reference sentences. BERT, introduced by Google AI in 2018, is fundamental in NLP. BERTScore has been shown to correlate better with human evaluations and outperforms other metrics in model selection, as it captures semantic similarity between sentences more effectively.

(https://plainenglish.io/community/evaluating-nlp-models-a-comprehensive-guide-to-rouge-bleu-meteor-and-bertscore-metrics-d0f1b1)

- Human evaluation:

Human judges evaluate aspects such as coherence, relevance, contextual understanding, empathy and conversational fluency, which automated metrics cannot fully capture.

To assess the quality of an LLM's output, evaluators can use different techniques such as the Likert scale (from 1 to 5) to assess its relevance, fluency or informativeness. In addition, to assess the accuracy of an LLM, evaluators can use data labeling techniques to identify incorrect statements and categorize them into specific types of errors such as factual inaccuracies, thematic deviations, nonsense answers, etc.

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.