Q146. Which tool is primarily used to collect user feedback and annotations to improve RAG pipelines?

A. HuggingFace Trainer
B. OpenAI Fine-tuning API
C. Argilla ✅
D. FastAPI

Q147. LoRA and PEFT techniques are mainly used for:

A. Data annotation
B. Lightweight fine-tuning ✅
C. Vector storage
D. Document indexing

Q148. Which fine-tuning technique is best when you want to adapt large models using fewer parameters?

A. Traditional fine-tuning
B. LoRA ✅
C. Argilla
D. PromptLayer

Q149. The HuggingFace Trainer is used for:

A. API deployment
B. Fine-tuning and training transformer models ✅
C. Annotation dashboards
D. Logging user prompts

Q150. Which API allows fine-tuning models like GPT-3.5 and GPT-4 directly?

A. HuggingFace Trainer
B. OpenAI Fine-tuning API ✅
C. FastAPI
D. LangSmith

Q151. In a RAG pipeline, Argilla annotations are primarily used for:

A. Automating embeddings
B. Evaluating retrieved documents ✅
C. Deploying interactive apps
D. Optimizing APIs

Q152. If you want to quickly deploy an interactive RAG-based web app, which framework pair should you choose?

A. HuggingFace + Argilla
B. FastAPI + Streamlit ✅
C. LoRA + Trulens
D. W&B + PromptLayer

Q153. What is the main advantage of using LoRA or PEFT over full fine-tuning?

A. They require less computation and fewer parameters ✅
B. They support only OpenAI models
C. They provide annotation dashboards
D. They replace embeddings

Q154. Which tool allows continuous improvement of RAG systems using feedback loops?

A. Argilla ✅
B. LangSmith
C. PromptLayer
D. HumanEval

Q155. The OpenAI Fine-tuning API can be used to:

A. Deploy apps directly
B. Customize GPT models with your own data ✅
C. Monitor hallucinations
D. Build embeddings

Q156. Which tool focuses on optimizing retrieved chunks and improving generated answers?

A. Argilla ✅
B. HuggingFace Trainer
C. FastAPI
D. PromptLayer

Q157. If you want to fine-tune a transformer model locally, the recommended tool is:

A. LangSmith
B. HuggingFace Trainer ✅
C. OpenAI Fine-tuning API
D. Trulens

Q158. Which of the following enables you to deploy RAG apps with a backend API + frontend dashboard?

A. FastAPI + Streamlit ✅
B. Argilla + LoRA
C. HuggingFace + HumanEval
D. LangSmith + PromptLayer

Q159. To improve RAG accuracy, a common approach is to:

A. Increase API latency
B. Fine-tune models using feedback ✅
C. Disable embeddings
D. Avoid vector databases

Q160. Which approach reduces hallucinations in generated answers?

A. Logging prompts only
B. Evaluating faithfulness with Argilla ✅
C. Removing embeddings
D. Using fewer documents

Q161. PEFT stands for:

A. Pretrained Embedding Fine-Tuning
B. Parameter-Efficient Fine-Tuning ✅
C. Predictive Embedding Framework Tool
D. Prompt Evaluation Framework Toolkit

Q162. To trace improvements after fine-tuning, which tool best integrates with experiment tracking?

A. HuggingFace Datasets
B. Weights & Biases (W&B) ✅
C. Argilla
D. PromptLayer

Q163. Which combination is best suited for a closed-loop feedback system in RAG?

A. Argilla + LoRA ✅
B. PromptLayer + FastAPI
C. LangSmith + Streamlit
D. W&B + HuggingFace Datasets

Q164. The primary benefit of using the OpenAI Fine-tuning API is:

A. Real-time document retrieval
B. Adapting GPT models for custom tasks ✅
C. Managing vector databases
D. Debugging LangChain pipelines

Q165. Which fine-tuning technique is most efficient when working with large language models on small GPUs?

A. PEFT ✅
B. Standard full fine-tuning
C. PromptLayer
D. Argilla

Q166. To create an evaluation + annotation dashboard for RAG improvement, the best choice is:

A. Argilla ✅
B. FastAPI
C. W&B
D. HuggingFace Trainer

Q167. For custom dataset fine-tuning on HuggingFace, which class is commonly used?

A. AutoEval
B. Trainer ✅
C. EmbeddingHandler
D. PromptDebugger

Q168. Which is NOT a fine-tuning tool?

A. LoRA
B. PEFT
C. HuggingFace Trainer
D. PromptLayer ✅

Q169. To continuously improve retrieval accuracy and response generation, you should:

A. Disable embeddings
B. Use Argilla feedback to fine-tune models ✅
C. Increase query size only
D. Avoid vector databases

Q170. Which combination provides an end-to-end fine-tuning + deployment pipeline for RAG apps?

A. HuggingFace Trainer + FastAPI + Streamlit ✅
B. PromptLayer + LangSmith + Argilla
C. W&B + Trulens + LoRA
D. HumanEval + OpenAI Playground