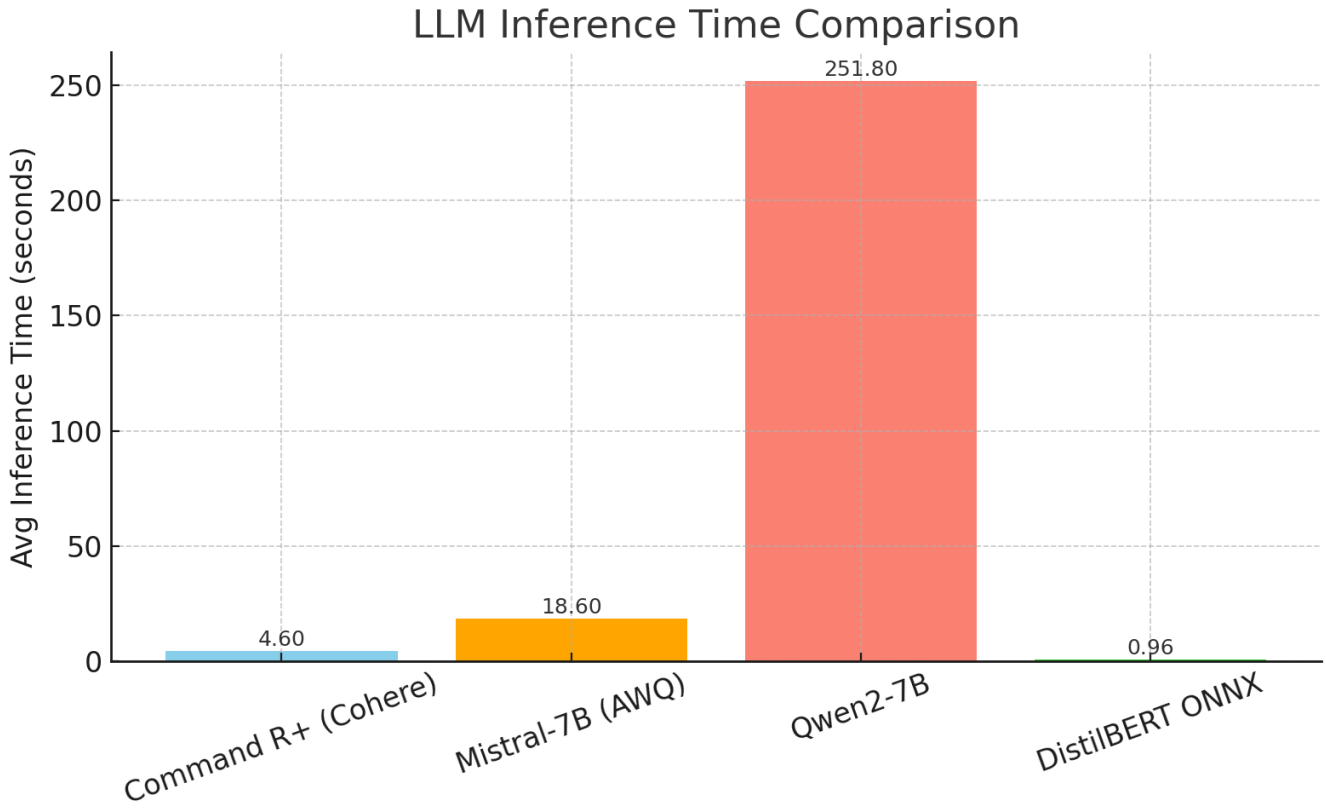
LLM Benchmarking Summary

- Command R+ (Cohere) provided the fastest hosted generation with strong context alignment.
- TheBloke/Mistral-7B-Instruct-v0.1-AWQ ran well using 4-bit AWQ quantization on Colab.
- Qwen/Qwen2-7B-Instruct produced rich and nuanced outputs, but with high latency.
- DistilBERT ONNX (distilbert-base-uncased-finetuned-sst-2-english) achieved sub-second inference on CPU with INT8 dynamic quantization.

This benchmark highlights trade-offs between inference latency, quality, and optimization strategies across hosted APIs, transformer-based models, and ONNX-optimized quantized models.



Benchmarking Methodology

All tests were conducted in Google Colab Pro using a T4 GPU runtime for transformer-based LLMs, and a CPU environment for ONNX-based quantized models.
Latency was measured using Python's time.time() function, capturing total response time from model prompt to final output.

Test Details:

- Prompt: "Explain the concept of model quantization in simple terms."
- Max Tokens: 150
- Sampling: do_sample=False (deterministic output)
- Timing Method: start = time.time() before generation, end = time.time() after
- Frameworks Used: transformers, cohere, optimum, onnxruntime
- Environment: Google Colab Pro (T4 GPU), CPU for ONNX
- ONNX Model Tooling: Hugging Face Optimum main_export()
- Quantization Types: 4-bit AWQ (Mistral), Dynamic INT8 (DistilBERT)