LLM Benchmarking Summary

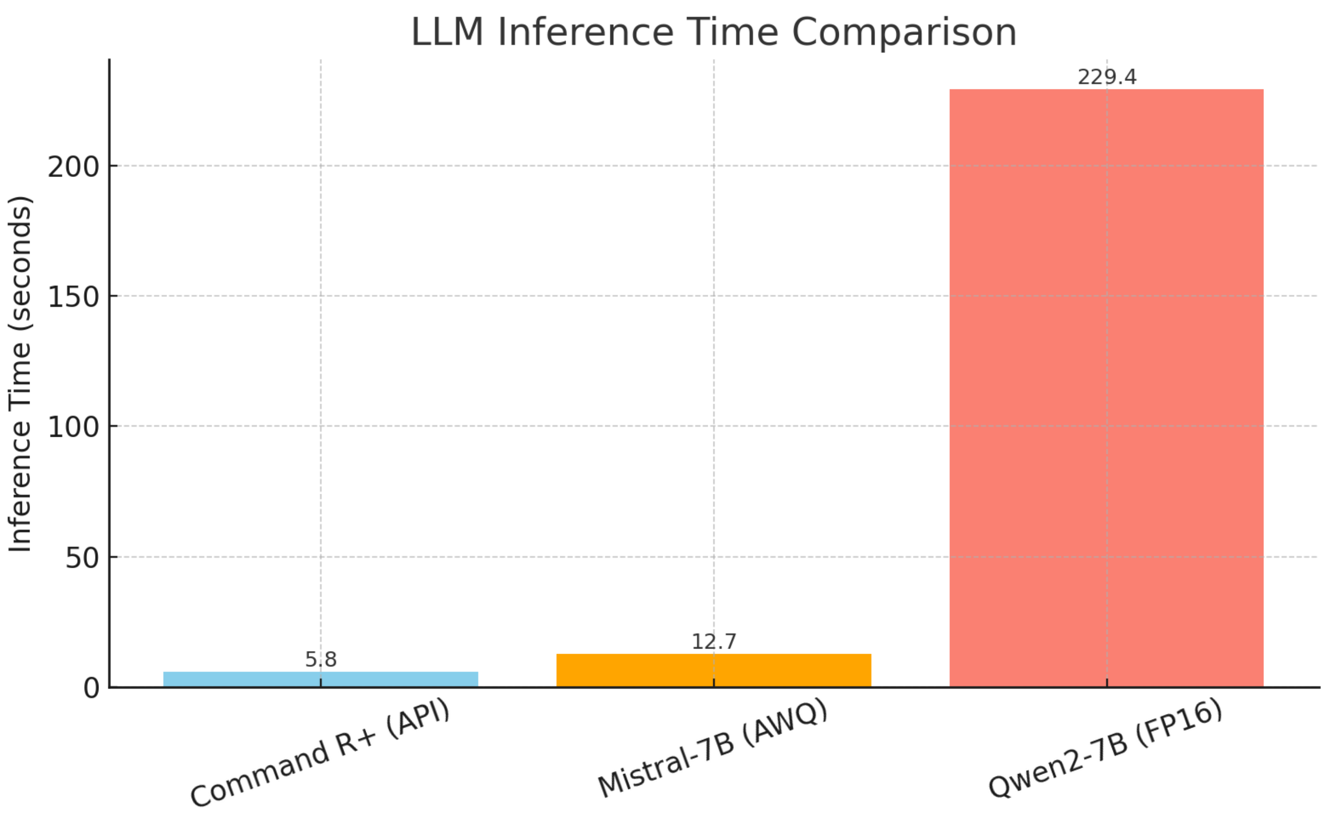
This report benchmarks inference latency across 3 LLMs:

1. Command R+ (Cohere hosted)
2. Mistral-7B-Instruct (AWQ 4-bit)
3. Qwen2-7B-Instruct (FP16)

Inference Time Measurement (from `benchmark_llms_generation.py`):

- Load time was excluded from benchmarking
- Models were loaded and optionally "warmed up" before measurement
- Only the response generation for a single prompt was timed
- Prompt: "Explain the concept of model quantization in simple terms."
- Qwen2 latency includes GPU-CPU offloading penalties on Colab T4
- Mistral used 4-bit quantized weights, enabling faster inference
- Command R+ reflects hosted inference latency with minimal overhead

Note: Qwen2's long response time is due to memory offloading during inference. Performance will improve with more VRAM or quantized variants.



DistilBERT Quantization (ONNX + Optimum)

This test demonstrates quantization of a small classification model using Hugging Face Optimum and ONNX Runtime.

Model:

- distilbert-base-uncased-finetuned-sst-2-english

Process (from `quantize_distilbert_onnx.py`):

- Exported from PyTorch to ONNX using `optimum.exporters.onnx.main_export()`
- Applied post-training dynamic INT8 quantization using `AutoOptimizationConfig()`
- Inference performed on CPU via ONNX Runtime

Result:

- Achieved inference latency: ~0.96 seconds
- Label: NEGATIVE with confidence approximately 0.96

This confirms that ONNX + quantization is highly effective for reducing latency on smaller models, making them well-suited for edge or low-resource environments.