7ZC-08

LLM サーバ構成を選択するための評価支援システムの提案

髙瀬 和真[†1]　太田 晶[†1]　小林 彰人[†1]　田中　雅人[†2]
日本工学院八王子専門学校[†1]　エイビット[†2]

1. はじめに

本研究では，ローカルで動作可能な LLM（Large Language Model：大規模言語モデル）および稼働させるサーバ構成を選定する際の評価ツールとして，複数の LLM やハードウェア構成を半自動的に構築して実行し，その実行状況を動画等で体感的に確認できるシステムの開発を目指している．

ChatGPT をはじめとした大規模言語モデル（LLM：Large Language Model）を用いたアプリケーションは，ビジネス分野においても開発や活用が進んできている．これら LLM は多くの場合クラウドサービスや API として提供されており，登録するだけで即時に利用可能な場合が多い．一方で，クラウドサービスならではの制約として，強制的な仕様変更や利用者増による応答速度の低下，また，組織におけるコンプライアンス上，投入できるデータが限られるなどの課題がある．

そこで注目されているのが，オープンソース等で開発・公開されており，オンプレミス環境やローカル環境で動作する LLM である．これらの LLM はその性能や要求されるハードウェア性能に大きなばらつきがあるため，LLM の性能を評価する際には，主に二つの側面が重要となる．一つは応答速度，つまりどれだけ迅速に結果を提供できるかという点である．もう一つは，生成された内容の正確さや信頼性である．市場には多数のベンチマークが存在し，これらの側面を定量的に評価するためのデータが提供されている．しかし，これらの数値のみを見て，LLM とそれを稼働させるハードウェアを含めた総合的な性能や適用可能性を評価することは極めて困難である．特に，開発に関わる技術者ではなく，最終的なユーザー層や，予算決定を行う部署において「実際に体感するとどのような違いがあるか」ということを，LLM のモデルやハードウェア選定時に容易に試すことができれば，総合的に満足度の高いシステムの構築に繋げられることが期待できる．

2. LLM の評価指標の調査と検討

大規模言語モデル（LLM）の性能評価に関しては，多様な研究が行われている．これらの研究は，一連の明確な基準に基づいて数値を算出し，LLM の効率性や有効性を定量的に測定している． まず，速度に関する評価指標に焦点を当てると，関連する研究では様々な指標が提案されている．例えば，[1]の研究では，1 分あたりの完了リクエスト数，最初のトークンまでの時間 (Time To First Token, TTFT)，トークン間レイテンシー (Inter-Token Latency, ITL)，エンドツーエンドの遅延，そして一般的なリクエストごとのコストが重要な評価基準とされている．また, [2]の研究では, TTFT, 出力トークンごとの時間 (Time Per Output Token, TPOT)，レイテンシー，そしてスループットが評価指標として挙げられている．次に，生成物の質に関する評価指標について考察する．ELYZA-tasks-100[3]のような研究では，特定のタスクに対するモデルの出力と期待される出力との間の類似性を測定する方法が提案されている．このアプローチでは，GPT-4 のような高度なモデルを使用して, 生成物に対して点数をつけ，その適切さを評価する．このような評価は，モデルの理解力や応答の質を，より実践的な観点から評価することを可能にしている．

これらの評価指標は，LLM の性能を多面的に理解する上で不可欠であり，それぞれがモデルの異なる側面を照らし出している．速度に関する指標は，実用的な応用における効率性を，生成物の質に関する指標は，モデルの応答の有用性や適切性を示している．本研究ではこれらの指標を総合的に考慮することで，LLM の全体的な性能評価を行うことができると考え，新しい評価支援システムを提案する．

3. 動画による LLM システム評価手法の提案

本研究では，ユーザーが直接視覚的に評価できる手法として，LLM の操作性と性能を比較検証するための基盤を構築することを提案する．この手法は，端末の種類や性能に依存せず，低コストで異なる計算リソースやモデル間のパフォーマンス比較を可能にすることを期待している．具体的には，LLM のタスク実行時の振る舞いや回答速度を，動画として記録し，これらを比較することで，ユーザーにとって直感的な理解を促すことを目的としている．

予備実験として構築したシステムの流れを以下に示す：

- ユーザーは, グラフィカルユーザーインターフェース（GUI）に優れた「Text generation web UI」[4]を通じて LLM を操作する．このステップでは，必要なモデルを事前にロードし，準備を整える．

A proposal of an evaluation support system for selecting LLM server configurations
†1 Kazuma TAKASE, Akira OHTA, Akihito KOBAYASHI
Nihon Kogakuin College of Hachioji
†2 Masato TANAKA, ABIT Corporation

- 「Playwright」[5]を用いた Python スクリプトにより，ブラウザ操作の自動化と動画出力を行う．このスクリプトは，web UI 上で行うべき一連の動作を定義し，それを順番に実行する．動画は 12 秒間記録した．
- スクリプトによる操作が完了した後，出力された動画をレビューし，その内容を評価する．

図 1 に予備実験のシステム構成を示す．予備実験では，LLM として ELYZA-japanese-Llama-2-7b を用い，GPU の違いによる推論結果を比較するため，より高性能な NVIDIA Tesla V100（VRAM:32GB）と，その 1 世代前の NVIDIA Tesla P100（VRAM:16GB）を用いた．また，生成するためのプロンプトは，ある程度共通の定量的な結果を期待し，かつ一般的なユーザーでも入力するような表現として「番号付き箇条書きで四字熟語を 100 個，1 番から生成してください」とした．

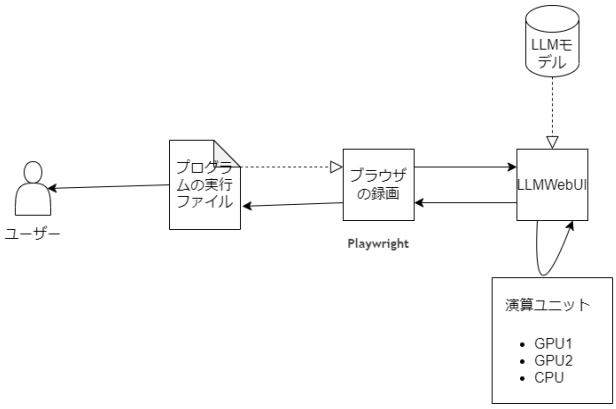

図 1　予備実験として構築したシステムの構成図

予備実験の結果として出力された動画のスクリーンショットを図 2 に示す．二つの GPU のうち高性能な V100 を用いた出力結果では，動画における 10 秒時点で 16 個目のテキストが表示されており，一方の GPU である P100 では，同じ 10 秒時点で 11 個目のテキストが表示されはじめた段階であった．箇条書きのテキスト 1 つあたりに出力されたテキストは高性能な V100 のほうが多いことから，明らかに出力速度が速いことが，体感的にも出力文字数からも確認できた．

一方で，出力内容の確からしさについては，GPU によらずプロンプトで指定した四字熟語という指定が守られておらず，これはプロンプトの工夫やモデルの選定やチューニングが必要と考えられる．

なお，体感的な速度を考察すると，性能の低い P100 の実行速度であっても実用上は問題ないと判断できるユーザーも十分にいると考えられ，このような動画を用いてユーザー対象の調査を行い，採用する LLM やハードウェア構成を選定することは，コスト削減にも繋がる可能性が示唆された．

図 2　LLM の実行状況を記録した動画（左 V100，右 P100）

4. まとめ

本稿では，ユーザーが LLM の操作に用いる Text generation web UI 上の操作を自動化し，またその様子を動画として出力する予備実験を行った．この予備実験により，LLM の評価手法で用いられる数値的な「速度に関する評価指標」や「生成物の質に関する評価指標」といった指標とは異なり，動画を視聴した人間の主観的な評価が容易になることを示した．また，NVIDIA Tesla V100 だけでなく，NVIDIA Tesla P100 など他のハードウェアにも容易に切り替えてテストすることができ，LLM の応答の速さを目視で体感的に確認できるようになった．その結果，求めるハードウェアスペックの選定が容易になることが期待される．

このシステムは，LLM の評価プロセスを容易にし，より多くのユーザーが専門的な知識を必要とせずに性能の違いを理解できるようにするための初歩的なステップとなる．本手法により，ユーザーは具体的な数値データや複雑なベンチマーク結果を解釈する代わりに，実際の動作を通じて直感的な理解を深めることができることが示唆された．

参考文献

[1] Waleed Kadous, Kyle Huang, Wendi Ding, Liguang Xie, Avnish Narayan, Ricky Xu “Reproducible Performance Metrics for LLM inference” https://www.anyscale.com/blog/reproducible-performance-metrics-for-llm-inference#additional-metrics-for-dedicated-instances , (参照 2024-01-07)

[2] Megha Agarwal, Asfandyar Qureshi, Nikhil Sardana, Linden Li, Julian Quevedo, Daya Khudia, ” LLM Inference Performance Engineering: Best Practices”, https://www.databricks.com/blog/llm-inference-performance-engineering-best-practices , (参照 2024-01-07)

[3] 中村, 佐々木, 堀江, 平川, “ELYZA が公開した日本語 LLM「ELYZA-japanese-Llama-2-7b」についての解説 : (2) 評価編” https://zenn.dev/elyza/articles/5e7d9373c32a98, (参照 2024-01-12)

[4] oobabooga, “Text generation web UI”, https://github.com/oobabooga/text-generation-webui?tab=readme-ov-file, (参照 2024-01-12)

[5] Microsoft, “Playwright for Python”, https://playwright.dev/python/, (参照 2024-01-12)