TECHNICAL PAPER — 株式会社喋ラボ

日本語ビジネス文書タスクにおける LLM量子化方式の比較評価

A Comparative Evaluation of LLM Quantization Formats for Japanese Business Document Tasks: Speed, Quality, and Memory Efficiency of GGUF Q4_K_M vs AWQ INT4

著者　大橋 功**所属**　株式会社喋ラボ**公開日**　2026年6月

ABSTRACT

本稿では、NVIDIA RTX 4060 Laptop（VRAM 8GB）という制約されたGPU環境において、オープンソースLLMの量子化方式（GGUF Q4_K_M および AWQ INT4）が日本語ビジネス文書タスクの速度・品質・メモリ使用量に与える影響を実測により評価した。対象タスクは日本語会議録を用いた（1）要約生成と（2）アクションアイテムのJSON構造化抽出の2種類とし、コンテキスト長（8K / 16K / 32K トークン）を変数として系統的に測定した。主要な発見として、GGUF Q4_K_M においてJSON有効率は全モデル・全コンテキストで100%を達成し、AWQ INT4（vLLM）は最小実験条件（8Kコンテキスト）を含む全条件で起動不可であることを定量的に示した。これらの結果は、8GB VRAM環境における実務的な量子化方式選択に具体的な指針を提供する。

1. はじめに

ローカル環境でのLLM実行に対する需要が急速に高まっている。プライバシー保護、レイテンシ削減、クラウドAPIコストの回避といった実務的要件から、企業が自社インフラ上でオープンソースLLMを運用するケースが増えている。

現在主流の量子化フォーマットには、汎用性の高いGGUF（llama.cppエコシステム）とNVIDIA GPU向けに最適化されたAWQ INT4がある。しかし既存の比較研究の多くは英語の汎用ベンチマーク（MMLU、HumanEvalなど）を用いており、**日本語のビジネス文書処理という実務タスクにおいて、VRAM 8GBという制約環境では何が現実的に動作するのか**という問いに答えていない。

1.1 本稿の主な貢献

- 日本語会議録を用いた要約・JSON抽出タスクでの量子化フォーマット比較
- コンテキスト長（8K/16K/32K）を変数としたGPUオフロード挙動の定量化
- AWQ INT4 + vLLMが8GB VRAM環境において最小実験条件すら起動不可であることの実測検証
- 実務者向けの量子化選択指針の提示

2. 背景と関連研究

2.1 量子化の基礎

LLMの重みはFP16（16ビット浮動小数点）で表現されると1パラメータあたり2バイトを要する。7〜8Bモデルのフルプレシジョン保存には約14GBが必要であり、VRAM 8GBのコンシューマーGPUには収まらない。量子化とは重みの数値表現を低ビット精度に変換する技術で、4ビット整数（INT4）への変換ではモデルサイズを約1/4に圧縮できる。

2.2 GGUF Q4_K_M

GGUFはllama.cppプロジェクトが開発した量子化フォーマットで、CPU・GPU・Apple Siliconのいずれでも動作する汎用性が特徴である。K-Quant（Q4_K_M）は重みをブロック単位で量子化し、重要な層を混合精度で保持する。VRAMが不足する場合、CPU/GPUハイブリッド実行が可能だが、PCIeバスがボトルネックとなる。

2.3 AWQ INT4

AWQ（Activation-aware Weight Quantization）は、各重みの出力への寄与を事前計測し、重要な重みを高精度で保持しつつそれ以外を積極的に圧縮する手法である。NVIDIA GPU向けの行列演算カーネルに最適化されており推論スループットが高い一方、vLLMはKVキャッシュをGPUメモリにあらかじめ確保する設計上、VRAM消費が大きい。

2.4 コンテキスト長とKVキャッシュ

8Bクラスのモデルでは、KVキャッシュは8Kコンテキストで約1.1GiB、16Kで約2.25GiB、32Kで約4.5GiBのVRAMを占有する。VRAM 8GB環境では、このKVキャッシュとモデル重みの競合が実験設計上の重要な変数となる。

3. 実験設計

3.1 ハードウェア

項目	仕様
GPU	NVIDIA RTX 4060 Laptop GPU
VRAM	8GB（8,188 MiB）
CUDA	12.8
ドライバ	580系
OS	Ubuntu Linux
ランタイム	llama-cpp-python 0.3.26（CUDA cu124）/ vLLM 0.22.1

3.2 評価モデル

モデル	パラメータ	アーキテクチャ	重みサイズ
Qwen3-8B	8B（Dense）	36層	約5.0GB
Gemma4-E4B	4B Effective（MoE）	42層	約5.4GB

3.3 タスク設計

タスクA：日本語会議録の要約

架空の日本語会議トランスクリプトを入力として200〜300字程度の要約文を生成させた。

タスクB：アクションアイテムのJSON構造化抽出

同じトランスクリプトから、担当者・タスク・期日・優先度を含む構造化JSONを出力させた。スキーマは以下のとおり。

```
{
  "action_items": [
    {
      "task": "タスクの内容（文字列）",
      "assignee": "担当者名（不明な場合はnull）",
      "due_date": "YYYY-MM-DD形式（不明な場合はnull）",
      "priority": "high / medium / low のいずれか"
    }
  ]
}
```

両タスクともInstruct用チャットテンプレートを適用し、Qwen3のthinking modeはオフに設定した。

3.4 テストデータ

データID	会議種別	トークン数	アクション数
T1	週次進捗MTG	2,591	7
T2	月次レビュー	6,515	12
T3	四半期計画MTG	12,681	20

全データは架空の日本語会議録であり、実在する会議・人物・組織とは一切関係がない。担当者・期日が明確なアクションアイテムと曖昧なもの（`due_date: null`）を意図的に混在させた。

3.5 実験条件マトリクス

モデル	量子化	8K	16K	32K	試行数
Qwen3-8B	GGUF Q4_K_M	✓	✓	✓	12
Gemma4-E4B	GGUF Q4_K_M	✓	✓	✓	12
Qwen3-8B	AWQ INT4	**ERROR**	**ERROR**	**ERROR**	3（全失敗）

4. 実験結果

4.1 GGUF Q4_K_M：速度・メモリ測定結果

Qwen3-8B（Q4_K_M）

コンテキスト	tok/s	TTFT（秒）	ピークVRAM	GPUオフロード
8K	**28.6**	2.5	6.62 GB	36/36層（全層）
16K	21.7	6.3	7.69 GB	33/36層（部分）
32K	8.0	18.4	7.46 GB	18/36層（部分）

8Kでは全36層をGPUに載せ28.6 tok/sを実現。32Kでは半数以上がCPUオフロードとなり、PCIeバス転送がボトルネックとなって速度は8.0 tok/sまで低下、TTFTも18.4秒に増加した。

Gemma4-E4B（Q4_K_M）

コンテキスト	tok/s	TTFT（秒）	ピークVRAM	GPUオフロード
8K	**41.0**	1.3	4.50 GB	42/42層（全層）
16K	38.5	2.4	5.11 GB	42/42層（全層）
32K	35.8	4.2	6.49 GB	42/42層（全層）

MoEアーキテクチャの特性によりKVキャッシュが小さく、32Kでも全42層をGPUに載せることができた。速度は8K（41.0 tok/s）から32K（35.8 tok/s）で13%しか低下しなかった。

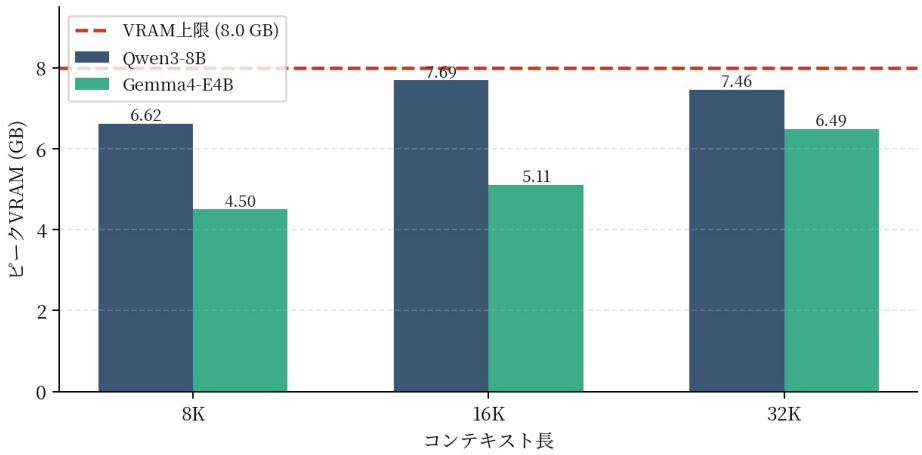

図2　ピークVRAM使用量（GB）— コンテキスト長別

4.2 AWQ INT4：起動不可の定量記録

> **結論：AWQ INT4（Qwen3-8B-AWQ）は VRAM 8GBでは実験の最小条件（8Kコンテキスト）すら起動できない。**4bit重みでもGPU上で5.71 GiBを占有し、KVキャッシュに回せるのは最大0.73 GiB（util=0.97）のみ。8K必要量1.12 GiBに届かない。

gpu_memory_utilization	KV確保量	推定最大コンテキスト	8K起動
0.90	0.19 GiB	1,392 トークン	**失敗**
0.93	0.42 GiB	3,056 トークン	**失敗**
0.95	0.57 GiB	4,160 トークン	**失敗**
0.97	0.73 GiB	5,280 トークン	**失敗**

なお縮小条件（ctx=4096、util=0.97）では起動成功・推論完了（22.67 tok/s、有効JSON）を確認。問題は純粋なハードウェアのメモリ容量によるものである。

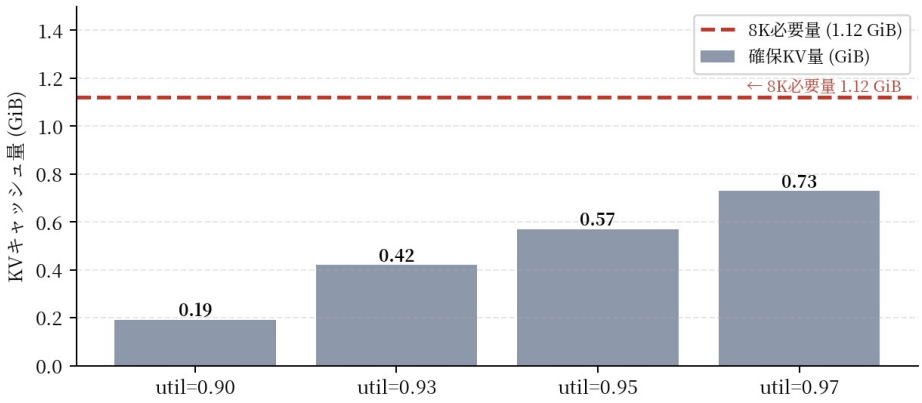

図5　AWQ INT4 — gpu_memory_utilization別KVキャッシュ確保量と8K必要量

4.3 品質比較：JSON抽出タスク

全GGUF試行（24試行）においてJSON有効率は**100%**であった。

モデル	8K	16K	32K	平均
Qwen3-8B	0.615	0.612	0.619	**0.615**
Gemma4-E4B	0.250	0.365	0.343	0.319

Qwen3-8BはF1スコア0.61〜0.62と安定して高品質な抽出を行い、コンテキスト長が変化してもF1スコアがほぼ一定であることが注目される。Gemma4-E4Bは個々のフィールド（担当者・期日・優先度）の抽出精度は1.0だが、アクションアイテム自体の網羅性（Recall）が低くF1が低下した。

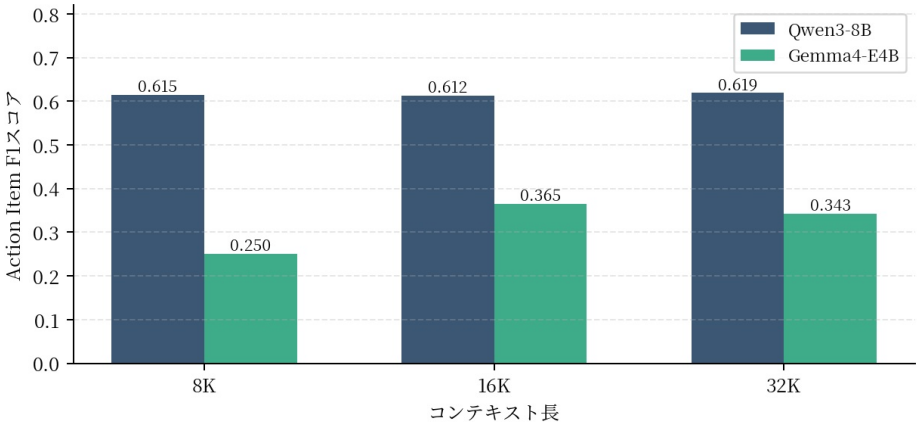

図3　アクションアイテム抽出 F1スコア — コンテキスト長別

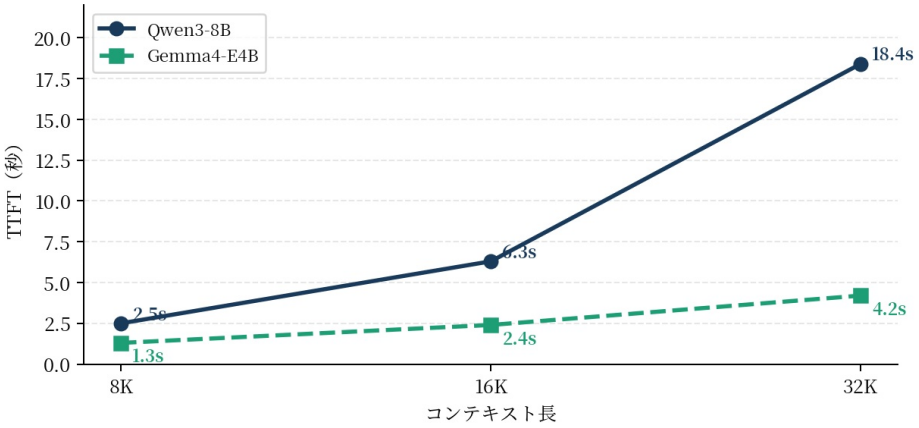

図4　TTFT（最初のトークンまでの時間、秒）— コンテキスト長別

4.4 速度・品質トレードオフの全体像

モデル	量子化	8K tok/s	32K tok/s	平均F1	32K VRAM
Qwen3-8B	GGUF Q4_K_M	28.6	8.0	**0.615**	7.46 GB
Gemma4-E4B	GGUF Q4_K_M	41.0	35.8	0.319	6.49 GB
Qwen3-8B	AWQ INT4	**起動不可**	**起動不可**	—	—

5. 考察

5.1 「AWQはGGUFより高品質」という通説の検証

AWQはGGUFより品質保持率が高い（AWQ 95% vs GGUF Q4_K_M 92%）と広く言われている。しかし本実験が示したのは、VRAM 8GBという制約環境ではAWQの品質優位性を発揮する以前に、そもそも実用的なコンテキスト長での起動が不可能だということである。

> 量子化方式の選択はモデル品質だけでなく、**ランタイムとハードウェアの組み合わせを含めたシステムレベルで評価すべき**である。

5.2 コンテキスト長の影響

Qwen3-8BにおいてGPUオフロード増加に伴う速度劣化が顕著であった（8K: 28.6 → 32K: 8.0 tok/s、3.6倍の差）。一方でF1スコアには有意な変化がなかった（0.612〜0.619）。**コンテキスト長の変化は品質に影響せず、速度とメモリにのみ影響する**という発見は実務上重要である。

5.3 モデルアーキテクチャとVRAM挙動

GemmaのMoEアーキテクチャが示したVRAM効率の高さ（32Kで全層GPUが可能）は、単純なパラメータ数でVRAMを見積もることの危険性を示している。活性化パラメータ数と注意機構の設計がVRAM消費を決定する。

5.4 実務的な選択指針

```
【VRAM 8GBでの量子化方式選択フロー】

コンテキスト長 < 16K かつ 速度優先？
  → Gemma4-E4B GGUF Q4_K_M（41 tok/s、4.5GB）

品質が重要かつ コンテキスト長 ≦ 8K？
  → Qwen3-8B GGUF Q4_K_M（28.6 tok/s、6.6GB）

長いコンテキスト（16〜32K）が必要かつ品質重視？
  → Qwen3-8B GGUF Q4_K_M（速度低下を許容）

AWQ INT4（vLLM）を使いたい？
  → VRAM 16GB以上が必要（8GBでは実用不可）
```

6. 限界と今後の課題

- **評価データの制約**：テストデータは架空の会議録であり、実際の業務会議録での性能を保証するものではない。
- **モデルのカバレッジ**：Qwen3-8BとGemma4-E4Bの2モデルに限定。他のモデルでは異なる結果が得られる可能性がある。
- **ランタイムの限定**：AWQ評価にvLLM 0.22.1を使用。他のAWQ対応ランタイム（ExLlamaV2など）では異なるメモリ効率を示す可能性がある。
- **今後の課題**：Mac mini（64GB統合メモリ）環境でのMLX 4bitとGGUF Q4_K_Mの比較、32B・MoEクラスモデルの評価を次フェーズとして計画している。

7. まとめ

本稿では、VRAM 8GBのGPU環境において日本語ビジネス文書タスクでの量子化方式比較を実施した。主要な知見を以下に整理する。

1. **AWQ INT4は8GB VRAMでは実用不可**：vLLM使用時、モデル重み5.71GiBによりKVキャッシュが最小実験条件にも不足し、全条件で起動失敗。
2. **GGUF Q4_K_M は全条件で完走、JSON有効率100%**：24試行すべてが正常完了。
3. **速度劣化はコンテキスト長依存、品質劣化はモデル依存**：Qwen3-8Bは32Kで速度が8.0 tok/sまで低下するがF1は安定（0.61）。Gemma4-E4Bは32Kでも35.8 tok/sを維持するがF1は0.32に留まる。
4. **アーキテクチャがVRAM挙動を決定する**：GemmaのMoE構造はKVキャッシュを削減し、32KでもVRAM 6.5GBで全層GPU実行が可能。

付録A：実験環境の詳細

項目	値
GPU	NVIDIA RTX 4060 Laptop GPU（8,188 MiB）
CUDA Toolkit	12.8
Python	3.11.15
llama-cpp-python	0.3.26（CUDA cu124）
vLLM	0.22.1

AWQ実験時の追加設定：`VLLM_WORKER_MULTIPROC_METHOD=spawn`（fork時のCUDA再初期化エラー回避）、`VLLM_USE_FLASHINFER_SAMPLER=0`（ネイティブサンプラー使用）

付録B：テストデータの統計

ID	会議種別	トークン数	アクション数	due_date未定
T1	週次進捗MTG	2,591	7	2
T2	月次レビュー	6,515	12	3
T3	四半期計画MTG	12,681	20	2

Citation

```
@techreport{ohashi2026quantization,
  title     = {日本語ビジネス文書タスクにおけるLLM量子化方式の比較評価},
  author    = {大橋 功},
  institution = {株式会社喋ラボ},
  year      = {2026},
  url       = {https://github.com/kouohhashi/llm-quantization-benchmark}
}
```