

RTX Spark 128GB環境で知財業務向けLocal LLMを構築するための国産LLM選定・注意点調査

エグゼクティブサマリー

結論として、128GB統合メモリを備えるNVIDIA RTX Spark搭載PCで知財業務用Local LLMを構築する場合、第一候補は **Qwen3-Swallow-32B**、第二候補は **Stockmark-2-100B-Instruct**、第三候補は **CyberAgentLM3-22B-Chat (CALM3)** とするのが妥当である。ただし、「国産」を何と定義するかで順位は変わる。Qwen3-Swallowは東京科学大学・産総研による国内開発だが基盤はAlibabaのQwen3であり、完全なフルスクラッチ国産ではない。一方、Stockmark-2、CALM3、LLM-jp、PLaMoなどは国内でゼロから学習した、またはそれに近い「ソブリン性」の高いモデルである。したがって、**性能・導入容易性を優先するならQwen3-Swallow、モデル由来を国内に閉じることを重視するならStockmark-2/CALM3**という二軸で判断すべきである。 ¹

RTX Sparkは最大128GBのCPU/GPU共有型統合メモリとBlackwell系GPUを備え、NVIDIAは同系統のSparkプラットフォームについてローカルで大規模モデルを扱うことを主要用途に掲げている。重要なのは、この「128GB」は一般的なPCの「RAM 128GB + GPU VRAM」とは異なり、**OS、モデル重み、CUDAランタイム、KV cache、RAG、アプリケーションが同じ有限なメモリ空間を競合すること**である。したがって本報告では、128GBを丸ごとモデルに割り当てず、実運用では概ね100～112GB以下をLLM関連処理の安全圏と考える。これはNVIDIAの公称値ではなく、本報告での運用設計上の余裕である。RTX Sparkの正確なSKU、TDP、OS、GPU/SoC数、メモリ帯域、追加GPUの有無はユーザー情報からは**未指定**であり、性能値は単一RTX Sparkデバイスを前提にした計画値とする。 ²

知財用途では、単に「一番大きいLLM」を選ぶべきではない。特許・契約業務では、生成力よりも①**日本語の精密な読解**、②**長文中の用語・参照関係**、③**引用元の追跡可能性**、④**再現性**、⑤**機密保持**、⑥**ライセンス上の法務利用可否**、⑦**誤りを検出できるワークフロー**が重要である。特に先行技術調査はLLMの内部知識に依存してはならず、J-PlatPatなどの権威ある一次データをRAG/検索側で取得し、LLMを「検索語拡張・クレーン要素分解・文献対比・要約」に限定する設計が必要である。J-PlatPatは特許庁から案内される特許情報プラットフォームである。 ³

128GB一台における実用上のスイートスポットは22B～32Bの4-bitモデルである。 このクラスならモデル本体を20～35GB程度に収め、長めのコンテキスト、RAG、複数利用者、小規模QLoRAまで余裕を残せる。一方、Stockmark-2の100B級は4-bitで約65～80GB程度を見込み、batch=1中心なら十分現実的だが、長文・並列処理・微調整は制約される。LLM-jp-3-172Bは4-bitでも110～125GB程度に達する可能性があり、「**ロードできる可能性**」と「**知財業務で安定運用できる**」は別問題である。単一128GBでは原則として研究・比較用と位置付けるべきである。

さらに、モデルのライセンスは「商用可/不可」だけで判断してはいけない。例えばLLM-jp-3-172B-Instruct3は商用利用自体を許容し得る一方、利用規約に無資格者による法的サービス等への制限があり、知財・契約レビューを外部顧客向けサービスとして提供する場合は特に慎重な確認が必要である。また、同規約は輸出管理法令への遵守も明示している。 ⁴

したがって推奨構成は、**Qwen3-Swallow-32B/AWQ 4-bitを通常業務の主モデルにし、Stockmark-2-100B-Instruct 4-bitを難案件・第二意見用に評価し、CALM3-22B-Chatを高速・安定・フルスクラッチ国産の**

フォールバックとして保持する構成である。モデルを同時常駐させるより、モデルサーバーを切り替えるか、2台目のRTX Spark系ノードを追加するほうが運用は安定する。

前提、評価軸、そして「国産」の定義

RTX Spark 128GBというハードウェアの意味

RTX Sparkは2026年に発表されたArm系CPUとBlackwell GPUを統合するPCプラットフォームで、最大128GBの統合メモリとFP4 AI処理能力をローカルAI用途の主要機能としている。NVIDIAのDGX Sparkも現在「RTX Spark powered」と位置付けられ、128GB統合メモリで大規模モデルのローカル推論を想定している。⁵

したがって、従来の「GPU VRAM 24GBなので残りはCPU RAMへoffload」という考え方とは少し異なる。RTX SparkではCPUとGPUが統合メモリを共有するため、CPU offloadによって容量が増えるわけではない。一方で、ディスクリットGPUのPCIe越しoffloadに比べてメモリ管理上は柔軟になり得る。このため最優先すべき最適化は、CPU offloadよりも **重み量子化、KV cache削減、コンテキスト制御、Paged Attention、適切なbatch制御** である。

ユーザーの「RTX Spark-equipped PC」に搭載されているRTX Sparkの**正確なSKU、GPU相当部のコア数、電力枠、OS、メモリ帯域、RTX Sparkデバイス数は未指定**である。現在のRTX Spark製品群はPC形態・電力枠が複数あり得るため、本報告のtokens/sは製品ベンチマークではなく、モデルサイズからみた**容量・帯域律速を前提とする計画値**であり、実機値には±2倍程度の幅を見込むべきである。⁶

「国産LLM」を二種類に分けるべき理由

本報告では、実務上のリスク評価のため次の二群を区別する。

国内フルスクラッチ型は、日本企業・研究機関が基盤モデル自体をゼロから事前学習したものを指す。Stockmark-2は約2兆tokenで学習された100B級モデル、CALM3は2兆tokenからスクラッチ学習された22Bモデル、LLM-jp-3は国内プロジェクトによる172Bモデルである。PLaMoもPreferred Networksがフルスクラッチ系国産LLMとして開発を続けている。⁷

国内適応型は、海外基盤モデルを日本国内で継続事前学習・SFT・RL等したものを指す。Qwen3-SwallowはQwen3を基盤として東京科学大学・産総研が日本語能力を強化したモデルであり、RakutenAI-2.0-8x7B-InstructのHugging Face系譜にはMistral-7Bが基底として明記されている。⁸

これは単なる呼称の問題ではない。知財部門では、基盤モデルの出自、上流ライセンス、学習データに関する説明可能性、海外依存、モデル更新の統制が調達要件になり得るからである。一方で、「国内フルスクラッチ」だから著作権・訓練データリスクがゼロになるわけでもない。文化庁もAIと著作権について、学習段階と生成利用段階を区別し、個々のケースについて法的評価が必要であるとの整理を公表している。⁹

メモリ推定の読み方

以下の容量はベンダー公称動作保証ではなく、本報告のエンジニアリング推定である。概算の出発点は、

- BF16重み：約2 byte/parameter
- 8-bit重み：約1 byte/parameter
- 4-bit重み：約0.5 byte/parameter

で、さらにscale/metadata、CUDA workspace、attention buffer、KV cache、tokenizer、アプリケーション用メモリを加算した。したがって「100B × 0.5 = 50GBだから50GBで動く」という計算は危険であり、実運用では概ね10〜30GB程度の余白を確保すべきである。

fine-tuningについては、通常のAdam系フルfine-tuningでは重み以外にgradient・optimizer state・activationが必要なため、22B以上は128GB一台では事実上対象外である。以下の「FT」は主として**QLoRA/LoRA**を意味する。知財用途では事実知識をfine-tuningで焼き込むより、最新特許・契約・社内規程をRAGで取得し、fine-tuningは出力様式や分類規則に限定したほうが監査可能性が高い。

候補国産LLMの比較

モデル、ライセンス、日本語性能、アラインメントの比較

モデル	開発者・系譜	ライセンス／商用	規模	Tokenizer	日本語性能・公開ベンチマーク	Instruction / alignment
Qwen3-Swallow-32B	東京科学大学＋産総研。Qwen3を国内で日本語CPT/SFT/RLした国内適応型	Apache-2.0、商用可。公式にAWQ INT4も公開 ¹⁰	32B dense	Qwen3互換BPE系	公式評価で32Bの日本語平均約0.609、Japanese MT-Bench約0.894。異なるベンチ間の単純比較は不可 ¹⁰	CPT＋SFT＋RL。推論・指令追従向け
Stockmark-2-100B-Instruct	Stockmark。約2兆token、100B級をフルスクラッチ。日本語・業務文書重視 ¹¹	商用利用可能な公開モデルとして提供。配布経路がHF/NVIDIA NIMの場合で条項が異なり得るため取得時の利用規約保存が必要 ¹²	HF表示約96B、100B級 ¹³	独自/AutoTokenizer。公式カードでアルゴリズム名称が明示されない場合はartifactで確認	日本語・ビジネス知識を重点評価。特許専用公開ベンチマークは確認できないため社内IP評価が必須 ¹⁴	SFT＋DPO。日本語synthetic dataで後学習 ¹⁵

モデル	開発者・系譜	ライセンス／商用	規模	Tokenizer	日本語性能・公開ベンチマーク	Instruction / alignment
CyberAgentLM3-22B-Chat / CALM3	サイバーエージェント。2兆tokenからフルスクラッチ事前学習 ¹⁶	Apache-2.0、商用可 ¹⁶	22B表記、HF実表示約23B	AutoTokenizerによる独自tokenizer。ChatML prompt	CyberAgentはNejumi LLM Leaderboard 3でLlama-3-70B-Instructに匹敵と報告。2024年時点の評価である点に注意 ¹⁷	Chat fine-tune済み、ChatML ¹⁶
RakutenAI-2.0-8x7B-Instruct	楽天。Mistral系譜を基盤にしたMoE国内適応型。2 experts active ¹⁸	Apache-2.0、商用可 ¹⁸	47B total / 約13B active	Mistral系tokenizer	Japanese MT-Bench 7.08±0.035 。同カードではCALM3 6.93。ただし楽天自身がbenchmark contaminationと単純順位付けに注意を明記 ¹⁸	Instruct fine-tune済み
LLM-jp-3-172B-Instruct3	国立情報学研究所LLM-jp。172B、国内大規模研究プロジェクト ¹⁹	商用利用を含む利用が申請・規約下で可能。ただし 再配布禁止 など独自条件あり ²⁰	約172B	LLM-jp独自tokenizer	g-leaderboard系でInstruct3 overall約0.669、日本語生成約0.757等を報告 ¹⁹	複数instruction dataset、AnswerCareful等を利用 ¹⁹

モデル	開発者・系譜	ライセンス／商用	規模	Tokenizer	日本語性能・公開ベンチマーク	Instruction / alignment
PLaMo 3.0 Prime	Preferred Networks。国産フルスクラッチ系。2026年にPrime 3.0を展開 ²¹	API/オンプレミス等の商用提供。公開weight向け一般ライセンスとは異なり契約型	パラメータ数は確認した公開情報では非公表	PLaMo系独自。詳細は契約artifact確認	PFNはPLaMo 2.2 Prime等でJMLE・MedRECTを含む日本語評価改善を報告。3.0は推論/非推論モード、最大256k contextを訴求 ²²	商用品としてalignment済み
Sarashina2.2-3B-Instruct	SB Intuitions。日本語特化の小型フルスクラッチ系列	MIT、商用可 ²³	3B	Sarashina系AutoTokenizer	同サイズ帯で日本語最高水準と開発元が報告 ²³	post-training済みInstruct

なおSarashinaには70Bの事前学習モデルも存在するが、SB Intuitions自身が当初公開したSarashina2系列について「指示チューニングを施しておらず、そのまま実用サービスに供する想定ではない」と説明している。知財アシスタントの主モデルとしては、パラメータ数だけを理由にpretrained-only 70Bを採用するより、小型でもInstructモデルか、別系統の32B/100B Instructを優先すべきである。 ²⁴

ローカル実装、メモリ、量子化、セキュリティ、用途の比較

下表のメモリ値はRTX Spark実測ではなく、重み＋一般的なruntime/KV余裕を含む設計時の概算レンジである。「VRAM」はRTX Sparkの場合、実質的にはGPUから利用可能な統合メモリを意味する。

モデル	推論時の 概算メモリ	Fine- tuning概 算	8/4-bit	Docker / vLLM / ONNX / ggml・ llama.cpp成熟 度	RTX Spark 128GB 単体	オンブ レ・ privacy	知財用途 適合
Qwen3- Swallow-32B	BF16 約 72– 88GB / INT8 40– 50GB / 4-bit 24– 35GB	Full FT >400GB 級 / QLoRA 35– 60GB	◎ AWQ INT4公式 。 8-bitも一般 的HF量子化 が可能。 GPTQは公開 停止履歴に注 意 ¹⁰	高。HF/Qwen 系。Docker/ vLLM系に載せ やすい。ONNX は32Bでは主経 路に非推奨。 llama.cppは変 換artifactを実 機確認	◎ 最も 扱いや すい	完全 offline なら外 部保持 なし。デ ータ保 持はモ デルで なく運 用側の 設定	特許分析 ◎、先行 技術対比 ◎、claim draft ◎、契約 ◎
Stockmark-2-100B- Instruct	BF16 >200GB / INT8 約 110– 125GB / 4-bit 65– 80GB	Full FT >1TB / QLoRA概 ね85– 120GB超 で限界	beta版には AWQ artifact が公開されて いるが、 final版と同 一とみなさな い ¹³	高～中。 Transformers/ vLLM、 Docker、 NVIDIA NIM経 路あり ²⁵ 。 llama.cpp/ ONNXは主経路 ではない	○ 4- bit + batch1 推奨	ローカ ルHFな ら高。 NIMを 使う場 合は NIM側 利用条 件・通 信設定 を別確 認	特許分析 ◎、先行 技術対比 ◎、claim ◎、契約 ◎。技 術・業務 文書向け 有力
CALM3-22B-Chat	BF16 52– 65GB / INT8 28– 35GB / 4-bit 17– 25GB	Full FT >300GB / QLoRA 28–50GB	○ bitsandbytes 等で8/4-bit 化可能。 first-party量 子artifactは 別確認	高。HFカード が Transformers、 vLLM、 SGLang、 Docker利用例 を明示 ¹⁶ 。 llama.cppは変 換テスト要	◎	offline 運用容 易	特許分析 ○～◎、 検索 query生 成 ◎、 claim ○、契約 ○～◎
RakutenAI-2.0-8x7B- Instruct	BF16 約 105– 120GB / INT8 58– 70GB / 4-bit 34– 45GB	Full FT >600GB / QLoRA 50–80GB 程度	○。HFに quantization 派生も存在 ¹⁸	中～高。 Transformers確 実。MoE kernel、vLLM/ llama.cpp exact revision は受入試験要	○～ ◎、 4- bit推奨	offline 可	分析 ◎、 検索 ◎、 claim ○ ～◎、契 約 ○

モデル	推論時の 概算メモリ	Fine- tuning概 算	8/4-bit	Docker / vLLM / ONNX / ggml・ llama.cpp成熟 度	RTX Spark 128GB 単体	オンブ レ・ privacy	知財用途 適合
LLM-jp-3-172B-Instruct3	BF16 >360GB / INT8 >190GB / 4-bit 約 110- 125GB以上	Full FT >2TB / QLoRAも 128GB単 体では原 則非現実 的	技術的4-bit 化余地あり。 ただし派生 weightの再 配布禁止に注 意 ²⁰	HFカードは Transformers、 vLLM、 SGLang、 Dockerを案内 ²⁶ 。容量が最 大の障害	△ ロー ド可能 性はあるが運 用余裕が小さい	offline 可。た だし利 用規約 に追加 制約	分析能力 の研究価 値◎。本 番単一 Sparkで は非推奨
PLaMo 3.0 Prime	非公開の ため推定 不可	非公開	公開情報だけ では判定不可	ベンダー提供オ ンプレとしては 高いが、一般的 HF/llama.cpp 自己展開とは別 物	? ベン ダーに weight サイ ズ・対 応HW 確認必 須	オンブ レ提供 あり ²¹	商用日本 語業務◎ 候補。た だしDIY Local LLM選定 としては 情報不足
Sarashina2.2-3B-Instruct	BF16 約 8-12GB / 4-bit 約 4-8GB	Full FT 50-80GB 程度、 QLoRA <15GB目 安	○	中～高。 HF/ Transformers中 心	◎ 余裕	offline 可	文書分類 ◎、 keyword/ query生 成◎、高 度な claim/無 効資料評 価△

「Docker対応」はモデル固有機能ではなく、モデルを動かすruntimeをコンテナ化できるかという意味で評価している。また「ONNX対応」は「変換自体が理論的に可能」と「100B級をONNX Runtimeで効率よく本番稼働できる」は異なる。本用途では、大型LLMについてONNXを第一選択にせず、**vLLM/SGLang、Transformers、TensorRT-LLM/NIM、またはGGUF/llama.cpp**のうち実機で安定する経路を選ぶのが合理的である。Stockmark-2についてはNVIDIA NIMの公式モデルページが存在する一方、NIMでの提供条件と元モデル配布条件を同一視すべきではない。 ¹⁵

RTX Spark 128GBでの実装可否と性能設計

単一RTX Sparkを前提にした優先順位

実運用での容量階層は明瞭である。

22～32B 4-bitではモデル本体が20～35GB程度であるため、RAG index、32k級以上の文脈、複数セッション、embedding/reranker、OCR、アプリを同一128GB PCで動かす余裕がある。知財部門で「担当者が対話しながら明細書・引用文献・Office Action・契約書を読む」という用途には最もバランスがよい。

47B MoE 4-bitも十分現実的である。RakutenAI-2.0-8x7Bは47B全weightを保持する一方、推論時のactive parametersは約13Bであることが公式カードに明記されているため、適切なMoE kernelが使えればdense 47Bより計算量を抑えられる可能性がある。ただし「active=13Bだから13Bモデルと同じVRAMで動く」と考えるのは誤りで、全expert weightの保持が必要である。 ¹⁸

100B 4-bitはRTX Spark 128GBの性能を活かす上位構成である。重みが約55～60GB、runtime・KVを含め65～80GB程度と見積もれば、batch=1、16k～32k程度の実用コンテキストは設計可能である。ただし長文を複数ユーザーが同時処理すると余裕が急速に減る。Stockmark-2はこの階層である。 ²⁷

172B 4-bitは境界領域である。単純なweight計算だけで100GB前後となり、scale・metadata・CUDA workspace・KV・OSを加えると128GBに極めて近づく。したがってLLM-jp-3-172Bは「128GBなら数字上入りそう」という理由だけで本番候補にすべきではない。短いcontext、batch=1、徹底した量子化、場合によっては一部offloadを前提にした評価用途とするべきである。

推定レイテンシとbatchの考え方

以下はRTX Spark上でこれらの個別モデルを測った公称benchmarkではない。RTX SparkのSKU/TDP、量子化kernel、Windows Arm/WSL/Linux環境、attention実装で大きく変わるため、PoC容量計画用の概算である。

モデル階層	推奨形式	batch	batch=1 decodeの計画値	実務上の位置付け
3B	BF16または4-bit	4-16	約25-60 token/s	大量分類、抽出、router
22B	AWQ/GPTQ/BnB 4-bit	1-4	約8-20 token/s	対話型知財アシスタント
32B	AWQ 4-bit	1-4	約6-15 token/s	最もバランスが良い
47B MoE	4-bit	1-3	約6-15 token/s、kernel依存大	高品質対話、MoE最適化前提
96-100B	4-bit	1中心	約2-5 token/s	難案件・第二意見
172B	4-bit	1	約1-2.5 token/s	比較・研究。日常対話には重い

長い特許明細書ではdecode速度以上に**time-to-first-token (prefill)** が効く。例えば100ページを毎回全文promptへ投入すると、32Bでも応答開始までの待ち時間が伸び、100Bではさらに顕著になる。したがって「128GBあるから100k tokenを毎回投入する」のではなく、請求項→段落→引用文献という階層型retrievalを行い、LLMへ渡す証拠を絞るべきである。

batchを2～4へ増やすとaggregate throughputは改善し得るが、一件あたりの待ち時間とKV cacheは増える。知財業務の対話処理はサーバー型チャットほど同時利用が多くないことが多いため、**100Bはbatch=1、32Bはbatch=1～2を標準、夜間一括分析のみbatchを上げる設計**が扱いやすい。

量子化の方針

本用途では「4-bitにすればよい」ではなく、**BF16または8-bitを基準モデルとして精度差を測ってから4-bitを承認する**べきである。

Qwen3-Swallowについては開発元がAWQ INT4版を提供しているため、第一選択をAWQとするのが合理的である。特に公式プロジェクトではGPTQ版に過度な反復が発見され公開停止された経緯がある。この種の劣化は一般ベンチマーク平均だけでは発見できず、claim draftingでは同じ限定句を重複する、先行文献対比では段落を反復する、といった実害につながる。 10

Stockmark-2についてはHF組織に100B betaのAWQ版が存在する一方、最終版とbeta版を同一と見なしてはいけな。finalモデルを自前で量子化する場合、BF16原weightだけで約200GBになるため、128GB Spark一台での量子化処理自体が推論より難しいことがある。別の大容量GPUサーバーで量子化し、ライセンス上許される範囲で完成artifactをSparkへ移す、または開発元の最終量子化artifactを使うほうが安全である。

13

複数GPU・複数RTX Sparkがある場合

ユーザー情報ではRTX Spark GPU/SoC数は指定されていないため、標準設計は一台・一GPU論理デバイスとした。

OEM PCが別のCUDA GPUを追加でき、runtimeが複数CUDA deviceとして認識するならtensor parallelismを検討できる。ただし、32B程度では通信オーバーヘッドのため単一デバイスのほうが単純かつ速い可能性が高い。100B以上ではTPの意義が大きくなる。

二台のSpark級マシンを使う場合、低速ネットワーク越しのtoken-by-token tensor parallelismは通信が律速になり得る。高速インターコネクトの保証がないRTX Spark PCでは、**一台を32B常駐、もう一台を100B常駐とし、案件単位でroutingする**ほうが、無理に一つのモデルを二台へ分割するより運用しやすい。

128GB×2で256GBがGPUから実効的に利用できる高速TP環境なら、100BのBF16や172Bの8-bitも容量上は現実性が高まる。ただしこれは「メモリが256GBある」だけでは不十分で、CUDA topology、interconnect帯域、vLLM/TensorRT-LLMの対応を実機確認する必要がある。

Arm/Windowsという見落としやすいリスク

RTX Spark PCはArm系Windowsプラットフォームとして展開されているため、x86/Linux向けLocal LLM手順をそのまま流用すると、CUDA extension、AWQ/GPTQ kernel、bitsandbytes、FlashAttention、vLLMなどでwheelやビルドの互換問題が発生し得る。RTX SparkのAI性能が高くても、**ソフトウェアスタックがArm64 Windowsに対応していなければ実効性能は出ない。** 28

したがって実装時には、第一候補として **Windows 11 Arm + WSL2/Linuxコンテナ**を検証し、OEMがLinuxを正式サポートする機種ならLinux環境も比較する。Arm64 CUDA extensionが不安定な場合のfallbackとして、CUDA対応でsource buildした `llama.cpp` /GGUF経路を用意するのがよい。ただし、モデルごとにGGUF変換可能性を実機で確認し、単に「Llama-likeだから動く」と仮定してはいけな。

知財業務への適合性と推奨アーキテクチャ

特許分析

LLMが最も得意なのは、「請求項を構成要件へ分解する」「明細書中の対応段落を拾う」「複数文献との差分を自然言語で説明する」「技術用語の同義語・上位概念・英訳を生成する」といった**文献を与えた上での分析**である。

逆に、「この発明には新規性がない」「この特許は無効である」とモデル内部知識だけで最終判断させるべきではない。特許情報の一次確認はJ-PlatPat等で実施し、LLMの回答には**公報番号、段落番号、claim番**

号、公開日などの証拠を必須出力させるべきである。J-PlatPatはINPITが提供し、特許庁からも公式検索先として案内されている。 3

この用途ではQwen3-Swallow-32BとStockmark-2-100Bが有力である。32Bで大量候補をscreeningし、重要文献だけ100Bで再分析する二段構成は、単一100Bで全件を処理するより効率がよい。

先行技術調査

先行技術調査ではLLMを検索エンジンとして使用せず、以下のように分離するべきである。

1. claimをatomic limitationへ分解する。
2. 日本語・英語の同義語、表記揺れ、機能表現を生成する。
3. lexical/BM25検索とembedding検索を併用する。
4. rerankerで候補を絞る。
5. LLMに「文献に記載されている事項だけ」をclaim chartへ転記させる。
6. 公報原文を機械的に再取得し、引用が実在することを検証する。
7. 最終的な新規性・進歩性評価を知財担当者・弁理士が行う。

特に特許文献では「類似している」と「全ての構成要件が一文献に開示されている」は別問題であるため、semantic similarity scoreを法的結論に変換させてはいけない。

Claim drafting

Claim draftingでは大規模モデルの流暢さが逆に危険になる。LLMは自然な限定を容易に追加できるが、その限定に明細書上のsupportがなければ問題になる。そのため、claim生成と同時に必ず、

claim element → 明細書のsupporting paragraph → 図面符号 → support confidence

の対応表を生成させるべきである。

また機械検証で、少なくともantecedent basis、従属関係、同一用語の揺れ、数値範囲の変更、明細書に存在しない新語、引用符号の不整合を検出する。生成温度はextractive taskで0～0.1、draftingで0.1～0.3程度から評価するのがよい。創造性を高めるためtemperatureを上げることは、知財成果物では通常メリットより監査コストを増やす。

契約レビュー

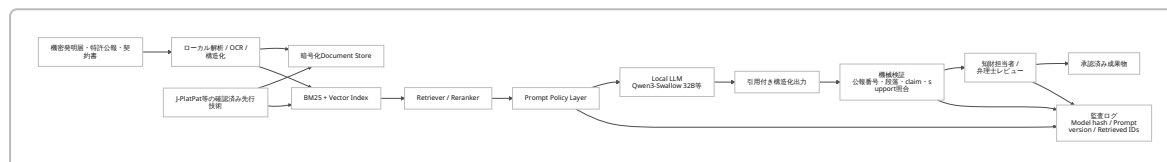
契約レビューでは、「契約書を丸ごとLLMに渡し法律判断をさせる」より、会社で承認済みの**IP clause playbook**をRAGにし、

- 知的財産権の帰属
- 背景IP/成果IP
- ライセンス範囲
- 再許諾
- 秘密保持
- 改良発明
- open source
- AI training/data use
- indemnity
- residual knowledge

などを構造化抽出し、playbookとの差分を提示させるのが安全である。

LLM-jpをこの領域で利用する場合は特に利用規約を確認する必要がある。LLM-jp-3-172B-Instruct3の規約には、無資格者による法律サービス・法的助言に関する禁止事項が含まれているため、社内補助用途と顧客向け法律サービスではリスク評価が異なる。 20

推奨するローカルRAG構成



この構成で重要なのは、「モデルが知っていること」と「案件について証拠から確認できたこと」を明確に分離することである。Local LLMの価値は、クラウドへ情報を送らないことだけではなく、モデル・retriever・prompt・データsnapshotを固定し、後から「なぜこの回答が出たか」を再現できる点にもある。

法務、機密、セキュリティ上の注意事項

日本のAIガバナンスについては、経済産業省・総務省の「AI事業者ガイドライン」が2026年3月31日に第1.2版へ更新されている。社内Local LLMであっても、「クラウドに送っていないからガバナンス不要」と考えるべきではなく、開発者・提供者・利用者としての責任分担、リスク管理、説明可能性、適切な運用統制を社内規程へ落とすべきである。 29

データ取扱いと機密保持

未公開発明、共同研究情報、NDA資料、license negotiation、M&A関連IP、係争資料は最上位の機密区分として扱うべきである。

Local LLMでは外部APIへpromptを送らない構成を取れるが、「Local = 漏えいしない」ではない。漏えい点はモデルAPI、Web UI、RAG database、embedding cache、prompt history、Windows/WSL filesystem、バックアップ、crash dump、管理者アカウント、ログ、ブラウザ拡張、OCRツールなどに移る。

個人情報保護委員会も生成AIサービス利用について個人情報入力時の注意を公表している。Local化後も、個人情報を必要以上にRAGへ投入しない、アクセス目的を限定する、ログ保持期間を決める、削除可能にする、といったデータ最小化を実施すべきである。 30

推奨する最低限の技術統制は、LLMサーバーのdefault-deny outbound firewall、disk encryption、OS account/SSO、role-based access、案件単位のRAG namespace、encrypted backup、管理者操作audit、model download用ネットワークと本番推論ネットワークの分離である。

モデルprovenanceとサプライチェーン

モデルを「Hugging Faceから一度ダウンロードしたもの」として管理せず、正式なソフトウェア構成品として管理する。

記録対象は、model repository、revision/commit、Safetensors各ファイルのSHA-256、tokenizer、chat template、license全文、NOTICE、quantization方法、calibration dataset、runtime version、CUDA/driver version、Docker image digestである。

本番環境では可能な限り `Safetensors` を使用し、`trust_remote_code=True` が必要なモデルではremote Python codeを事前reviewして社内artifactへ固定する。ネットワーク接続した状態で毎回 `latest` を取得する運用は監査再現性を失う。

ライセンス遵守

Apache-2.0やMITだから「何をしてもよい」わけではない。ライセンスは主としてモデルweight/codeの利用条件を定めるもので、入力文書、訓練データ、生成物が第三者著作権を侵害しないことを保証するものではない。AIと著作権に関する日本政府の整理でも、AI学習と生成・利用局面の法的評価は区別されている。⁹

特に以下をモデルごとに保存する。

Qwen3-Swallow/CALM3/RakutenAIではApache-2.0のLICENSE/NOTICEを保持する。Qwen3-SwallowとRakutenAIは海外base modelの系譜もSBOM相当資料へ含める。³¹

LLM-jpでは独自Terms of UseをApacheモデルと同様に扱ってはいけない。モデル・改変物の再配布制約、法律サービス制約、輸出管理条項を必ず確認する。量子化weightは技術的には改変物に当たり得るため、社内利用と社外配布を分離する。²⁰

PLaMo Primeでは一般OSSライセンスではなく商用契約条件を精査し、オンプレ権、複製可能台数、バックアップ、災害復旧、fine-tuning、adapter所有権、ログ、契約終了時のweight消去条件を確認する。PFNはPLaMo PrimeをAPI・オンプレを含む商用形態で提供している。²¹

Stockmark-2では元モデル取得経路とNVIDIA NIM利用経路を区別する。NIM側ページにはNVIDIA側termsが存在するため、「Stockmarkモデルが商用可だからNIMコンテナも同条件」と仮定してはいけない。¹⁵

著作権と生成物

特許公報自体を分析することと、モデルが別資料を長く再現することは別の問題である。生成物の採用前には、長い特徴的文章の再現、第三者契約条項の丸写し、参考文献からの過剰なコピーをチェックする。

特にclaim draftingでは、他社公報のclaimを「表現を少し変えて」生成することより、自社明細書の記載から構成要件を導出するprompt設計にする。文化庁のAI・著作権資料は法的拘束力そのものではないが、企業のAI著作権リスクを整理する際の重要な公的参照資料である。⁹

輸出管理

海外子会社へのmodel weight送付、海外からのremote access、海外拠点へのRTX Spark機器移設、暗号・AI関連softwareの提供は、通常の「国内PC利用」とは別に輸出管理レビューを行う。経済産業省は安全保障貿易管理として貨物・技術提供に関する制度を運用している。³²

特にLLM-jp利用規約自体が日本の外国為替及び外国貿易法や米国輸出管理等への対応を利用者側に求めているため、海外共有を「Hugging Faceで公開されているweightだから自由」と判断してはいけない。²⁰

Hallucination防止

知財LLMでは「hallucinationを減らす」だけでは不十分で、hallucinationが混入した場合に自動的に検出し、人が採用しない仕組みが必要である。

最重要ルールは、事実主張を原則としてretrieved evidenceへ拘束することである。モデルには「公報に明記がなければ“不明/記載なし”と答える」「引用段落なしに新規性結論を出さない」「公報番号を生成せず retrieverから渡されたIDのみ使用する」とsystem promptで指示する。

さらに回答後に、LLMとは独立したコードで公報番号・段落番号・claim番号の存在チェックを行う。**LLMに自分自身の引用を検証させるだけでは検証にならない。**

先行技術調査では次の三分類を強制するとよい。

ENTAILMENT : 引用文献が構成要件を明示的に開示

PARTIAL : 一部のみ開示

NOT FOUND : 取得文献内に根拠なし

「おそらく開示されている」「一般的に知られている」といった自由記述を最終claim chartへ残さない。

Prompt engineering

知財promptは、役割設定より**出力schemaと根拠制約**が重要である。

悪いpromptは「この発明の特許性を評価してください」。

良いpromptは「請求項1を最小の技術的限定へ分割し、各限定についてDocument Aの該当箇所を原文引用、段落番号、entailment/partial/not-foundの三値、理由を出力せよ。与えられたDocument A以外の知識を根拠にしてはならない」という形式である。

Claim draftingでも「良い請求項を書け」ではなく、「明細書の段落X～Yに明示又は直接支持される構成のみ使用」「新しい数値範囲は禁止」「追加した語句ごとにsupporting paragraphを返す」とする。

Loggingとaudit

最低限、以下を一つの「AI案件記録」として再現可能にする。

model-id + model hash + quantization hash + tokenizer hash + prompt-template version + retrieved document IDs + corpus snapshot date + generation parameters + output + automatic validation result + human reviewer + final decision

ただし、auditを理由に全prompt原文を永久保存すると、監査DB自体が新たな機密情報集積点になる。機密度に応じ、plaintext log、redacted log、content hashのみ、という三段階を使い分け、retention periodを明文化する。

知財精度のvalidation

一般LLM leaderboardだけで本番採用を決めてはいけない。Rakutenのモデルカード自身も、instruction tuningとbenchmark contaminationの関係から「高いLLM-JP評価値だけで他モデルより優秀と言い切るべきでない」と明示している。¹⁸

社内で少なくとも次のgolden setを作るべきである。

評価タスク	推奨指標
Claim要素分解	弁理士/担当者goldとのprecision・recall
公報段落抽出	Citation precision / citation recall
Claim chart	Element-level entailment F1
先行技術ranking	Recall@10、Recall@50
架空公報生成	Hallucinated citation rate
明細書support	Unsupported limitation rate
Claim drafting	antecedent basis error、dependency error、support error
契約レビュー	clause extraction F1、playbook issue recall
日英特許翻訳	用語一貫性＋専門家blind review
量子化劣化	BF16/8-bit/4-bitとの差分
再現性	同一入力での構造化結果一致率

本番合格基準は一般benchmarkではなく、例えば「citation hallucination 0件」「重大claim support error 0件」「先行文献Recall@50 98%以上」など、自社のrisk appetiteに合わせて決めるべきである。最終判定者はLLMではなく、知財担当者・弁理士・弁護士等の資格・職責を有する人間とする。

推奨トップ候補と実装チェックリスト

最優先候補：Qwen3-Swallow-32B

総合推奨度：最も高い。

理由は、32Bという128GB Sparkに対して余裕のあるサイズ、2026年時点での日本語・推論性能、Apache-2.0、公式AWQ INT4の存在、SFT/RL済みという条件が揃っているからである。東京科学大学・産総研による国内日本語強化モデルで、日本語benchmarkでも高い値を報告している。¹⁰

推奨構成は**AWQ INT4**、**batch=1~2**。モデル＋runtimeを30GB前後に抑えられれば、RAG、reranker、OCR、長文KV、Web UIを同一PCで動かす余裕が大きい。

弱点は、完全なフルスクラッチ国産ではなくQwen3に由来することと、量子化形式による挙動差である。公式プロジェクトでGPTQ版の反復問題が発見された事実は、むしろ「量子化後に知財golden setを再評価する必要性」を示す好例である。¹⁰

推奨用途：特許要約、claim decomposition、先行文献claim chart、Office Action分析、claim drafting補助、契約レビュー。通常の知財担当者向け対話LLMとして最適。

高品質・ソブリン候補：Stockmark-2-100B-Instruct

総合推奨度：高。ただし4-bit artifactを先に確立できることが条件。

Stockmark-2は100B級をスクラッチから学習し、日本語とビジネス利用を強く意識したモデルで、SFT/DPOも実施されている。約2兆tokenの事前学習と日本語post-trainingが公式モデル情報で説明されている。¹¹

100B級でありながら4-bitなら128GBへ十分収まる見込みがあり、「国産基盤で高能力」という点は知財・研究開発部門にとって非常に魅力的である。

一方で、batch、context、QLoRAの余裕は32Bより小さい。推論専用の**難案件モデル**として、32Bで候補を絞った後に重要案件だけ100Bへ通す構成が合理的である。

さらに、finalモデルとbeta AWQを混同してはいけない。HFにはStockmark-2-100B-Instruct-beta-AWQが存在するため、final 100Bで同等artifactが正式提供されているか、または自社で量子化可能かを導入時に確認する。 ¹³

推奨用途：長い技術文書比較、重要特許の第二分析、複雑なclaim chart、契約条項推論、技術調査報告書。

安定・軽量フルスクラッチ候補：CALM3-22B-Chat

総合推奨度：高。能力単独より運用安定性を評価。

CALM3はサイバーエージェントが2兆tokenからスクラッチ学習した22Bモデルで、Apache-2.0、Chat fine-tune済み、16k contextを公式カードで明示している。HFページにはTransformersだけでなくvLLM、SGLang、Dockerの具体的利用方法も掲載されており、候補中でもdeployment情報が分かりやすい。 ¹⁶

22B 4-bitなら20GB前後+runtimeで収まり、128GB Sparkでは極めて余裕がある。32Bより高速な大量処理用、または「海外base modelを利用しない」という調達要件がある場合の主モデルとして有力である。

弱点は2024年世代であり、最新32B reasoningモデルより複雑な推論で劣る可能性があること。そのため一般benchmarkではなく知財golden setで判断する。

推奨用途：特許・契約の一次処理、分類、query expansion、要約、entity抽出、claim分解、大量文書screening。

次点候補

RakutenAI-2.0-8x7B-InstructはJapanese MT-Bench 7.08と公開評価が明確で、47B total/13B activeのMoEが魅力である。Qwen3-SwallowよりMoE運用が複雑で、Mistral由来という点を許容できる組織には有力な比較対象になる。 ¹⁸

LLM-jp-3-172B-Instruct3は国内研究モデルとして重要だが、単一128GBでは4-bitでも容量余裕が乏しく、独自利用規約も知財サービスには注意を要する。研究比較または複数デバイス環境での採用候補とする。 ³³

PLaMo 3.0 Primeは日本語商用LLMとして非常に興味深いが、公開情報からparameter数・量子化・一般weight self-host条件を確定できないため、「RTX Sparkに自分でロードするモデル」という今回の要件では比較不能部分が残る。PFNとのオンプレ契約まで含めたエンタープライズ調達で再評価すべきである。 ²¹

実装チェックリスト

ハードウェア確認

- RTX Sparkの正確なSKU、128GBが統合メモリであること、TDP、SSD容量、冷却性能をinventory化する。
- NVMeは最低でもモデル複数世代、RAG index、snapshotを保持できる容量を確保する。100B BF16原weightを保管する場合はモデル一つで約200GBになる。

- 2台以上の場合、CUDA topologyとネットワークを測定してからtensor parallelismを判断する。
- 本番PCを一般Web閲覧・メール用PCと兼用せず、知財AIワークステーションとして分離する。

OS・software stack

推奨概念構成は、

```
Windows 11 Arm → WSL2/Linux または OEM-supported Linux → CUDA → Docker → vLLM/SGLang/Transformers
```

を第一経路にし、

```
CUDA build llama.cpp + GGUF
```

をfallbackとする。

CALM3については公式HFページでvLLM/SGLang/Docker利用例が示されている。Stockmark-2にはTransformers/vLLMやNVIDIA NIM経路があり、Qwen3-Swallowは公式AWQを利用できる。³⁴

Arm64上では、導入前に必ず次を実機CIで確認する。

```
torch.cuda.is_available()
```

、BF16、4-bit kernel、Flash/Paged Attention、AWQ load、長文prompt、Docker GPU passthrough、suspend/resume後のGPU状態、48時間連続推論、メモリリーク。

量子化手順

まず公式repositoryからモデルを取得し、revisionをpinする。LICENSE/Terms、model card、SHA-256を保存する。

次に、可能なモデルではBF16または8-bitをreferenceとしてgolden setを実行する。

その後、Qwen3-Swallowなら公式AWQ INT4を投入し、同じテストを再実行する。Stockmark finalなどfirst-party 4-bitがない場合は、より大きな量子化用machineを使用するか、正式量子artifactを待つ/調達する。

4-bit化後に見るべきなのはperplexityだけではない。**引用hallucination、claim element抜け、反復、数値転記、否定表現、日英用語、表形式JSONの破損**を個別測定する。

量子化artifactにも元モデルとは別のhashとversionを付与し、本番で勝手に再量子化しない。

RAG構築

最初から全社文書をvector DBへ入れない。まず公開特許＋匿名化した過去案件でPoCし、アクセス制御が確立した後に未公開発明を追加する。

keyword/BM25とembeddingをhybridにし、特許番号・IPC/CPC・出願人・発明者・日付などのmetadata filterをLLMより前段で処理する。

J-PlatPat等の一次情報を最終確認sourceとして扱う。³

文書をchunk化する際は単純な500 token固定分割ではなく、claim、段落番号、実施例、図面説明、citation という法的・文書構造を保持する。

評価試験

PoCでは最低でも各候補に同一の50～200案件を処理させる。

32B BF16/8-bit、32B AWQ4、CALM3 4-bit、Rakuten 4-bit、Stockmark 4-bitを、同一retrieval結果・同一promptでblind比較する。

一般benchmarkスコアより、次を採用判定の上位指標にする。

citation hallucination rate、unsupported claim rate、claim-element recall、prior-art Recall@k、expert correction time。

特に「専門家が修正するのに何分かかったか」は重要である。モデルAの文章が美しくても毎回引用確認に10分かかり、モデルBは簡素でも2分で検証できるなら、知財実務ではBのほうが高性能である。

本番リリース判定

モデル、量子化方式、prompt、retrieverのどれか一つでも変えた場合はregression testを再実行する。

モデル更新は **latest** 追従ではなく、

candidate → security review → IP golden evaluation → legal/license review → approval → production

というrelease gateを通す。

現在の日本政府AIガバナンス参照点としては、2026年3月31日公表のAI事業者ガイドライン第1.2版を社内AI規程のbaselineの一つとするのが妥当である。 ²⁹

最終的な採用構成としては、**一台だけならQwen3-Swallow-32B/AWQを主モデル**とし、フルスクラッチ国産要件が強ければ**CALM3-22B**へ切り替える。100B級を許容できる待ち時間と4-bit artifactが確立できた段階で**Stockmark-2-100B-Instructを高精度tierとして追加**する。この構成が、RTX Spark 128GBの容量を過度に使い切らず、知財業務で最重要となる機密性、証拠追跡性、法的統制、モデル更新時の再現性を最も両立しやすい。

¹ ⁸ ¹⁰ ³¹ Qwen3 Swallow — Swallow LLM

<https://swallow-llm.github.io/qwen3-swallow.ja.html>

² ⁶ Slim Laptops & Small Desktops | NVIDIA RTX Spark

https://www.nvidia.com/en-us/products/rtx-spark/?utm_source=chatgpt.com

³ 特許情報プラットフォーム | J-PlatPat [JPP]

https://www.j-platpat.inpit.go.jp/?utm_source=chatgpt.com

⁴ ²⁰ ³³ llm-jp/llm-jp-3-172b · Hugging Face

<https://huggingface.co/llm-jp/llm-jp-3-172b>

- 5 28 NVIDIA and Microsoft Reinvent Windows PCs for the Age ...
https://nvidianews.nvidia.com/news/nvidia-microsoft-windows-pcs-agents-rtx-spark?utm_source=chatgpt.com
- 7 11 12 15 25 Stockmark-2-100B-Instruct
https://docs.api.nvidia.com/nim/reference/stockmark-stockmark-2-100b-instruct?utm_source=chatgpt.com
- 9 AIと著作権について
https://www.bunka.go.jp/seisaku/chosakuken/aiandcopyright.html?utm_source=chatgpt.com
- 13 27 Stockmark Inc.
https://huggingface.co/stockmark?utm_source=chatgpt.com
- 14 日本発の「ソブリンAI」として独自開発の日本語LLM「...
https://stockmark.co.jp/r-and-d/topics/20250925/?utm_source=chatgpt.com
- 16 34 cyberagent/calm3-22b-chat · Hugging Face
<https://huggingface.co/cyberagent/calm3-22b-chat>
- 17 独自の日本語LLM（大規模言語モデル）のバージョン3を一般公開
https://www.cyberagent.co.jp/news/detail/id%3D30463?utm_source=chatgpt.com
- 18 Rakuten/RakutenAI-2.0-8x7B-instruct · Hugging Face
<https://huggingface.co/Rakuten/RakutenAI-2.0-8x7B-instruct>
- 19 LLM-jp-3 172B: オープンかつ日本語に強いGPT-3級大規模言語モデル | 国立情報学研究所 大規模言語モデル研究開発センター
<https://llmc.nii.ac.jp/topics/llm-jp-172b/>
- 21 22 国産生成AI基盤モデルPLaMo 3.0 Primeを正式リリース
https://www.preferred.jp/news/pr20260622?utm_source=chatgpt.com
- 23 Sarashina2.2-Instruct：コンパクトかつ性能の高い日本語 ...
https://www.sbintuitions.co.jp/blog/entry/2025/03/07/093143?utm_source=chatgpt.com
- 24 SB Intuitions、独自の日本語LLMを構築 アカデミアや産業界 ...
https://www.sbintuitions.co.jp/news/press/20240614_01/?utm_source=chatgpt.com
- 26 llm-jp/llm-jp-3-172b-instruct3 · Hugging Face
<https://huggingface.co/llm-jp/llm-jp-3-172b-instruct3>
- 29 AI事業者ガイドライン
https://www.meti.go.jp/shingikai/mono_info_service/ai_shakai_jisso/20260331_report.html?utm_source=chatgpt.com
- 30 生成AIサービスの利用に関する注意喚起等について
https://www.ppc.go.jp/news/careful_information/230602_AI_utilize_alert/?utm_source=chatgpt.com
- 32 安全保障貿易管理
https://www.meti.go.jp/policy/anpo/?utm_source=chatgpt.com