Japanese HR NIAH: 架空データを用いた日本語人事労務領域のロングコンテキスト性能評価ベンチマーク

久保田崇文
株式会社 SmartHR
takafumi.kubota@smarthr.co.jp

概要

ロングコンテキスト LLM の実用化が進む中，大量の文書から必要な情報を正確に検索・統合する能力の評価が重要となっている．特に人事労務ドメインでは，就業規則，勤怠データ，契約書など多様な文書を統合し，複雑な質問に答える必要がある．しかし，既存の NIAH ベンチマークは一般的なドメインでの評価に留まり，日本語や特定ドメインに特化した評価は不足している．本研究では，日本語人事労務ドメインに特化した NIAH ベンチマークを構築した．架空企業の従業員 60 名分のデータを含む合成データセットを「干し草の山（Haystack）」として作成し，単一データ検索，複数データ検索，時系列データ検索の 3 種類のタスクタイプを設計した．主要な 6 モデルを評価した結果，モデル間で性能差が確認され，特に時系列データ検索タスクで難易度が高いことが明らかになった．本研究は，日本語ドメイン特化ベンチマークの提供と，人事 SaaS 企業における LLM 実用化の指針を示す点で貢献する．

1 はじめに

近年，大規模言語モデル（LLM）のコンテキスト長が飛躍的に拡大し，100 万トークンを超えるコンテキストを処理できるモデルが登場している．これにより，大量の文書を一度に処理し，その中から必要な情報を検索・統合するタスクが現実的なものとなってきた．特に，企業の人事労務部門では，就業規則，給与テーブル，雇用契約書，勤怠ログ，1on1 面談記録など，多様な形式の文書を統合的に扱う必要がある．

しかし，ロングコンテキスト LLM の実用化を進める上で，その性能を適切に評価するベンチマークが必要である．Needle In A Haystack（NIAH）ベンチマーク [1] は，大量の文書（干し草の山）の中から特定の情報（針）を検索する能力を評価する手法として広く用いられている．しかし，既存のベンチマークは主に英語の一般的なドメインを対象としており，日本語や特定の実務ドメインに特化した評価は不足している．

本研究では，日本語人事労務ドメインに特化した NIAH ベンチマークを構築し，架空企業の従業員 60 名分のデータを含む合成データセットを「干し草の山（Haystack）」として作成した．単一データ検索，複数データ検索，時系列データ検索の 3 種類のタスクタイプを設計し，主要な 6 モデル（GPT-5, GPT-5-mini, Gemini 3 Pro, Gemini 3 Flash, Claude Sonnet 4.5, Claude Opus 4.5）を評価した．

なお，本研究で構築したベンチマークは右記の GitHub リポジトリで公開している．https://github.com/kufu/Japanese-HR-NIAH

2 関連研究

LLM のコンテキストウィンドウが数万から数百万トークンへと拡大する中，その長文処理能力を正確に測定する包括的なベンチマークの必要性が高まっている．L-Eval[2]，ZeroSCROLLS[3]，LongBench[4] などが提案されているが，これらは平均数千〜数万トークンに留まっており，数十万〜数百万トークンを扱える最新モデルの限界を測るには不十分である．

一方，大量のテキスト（Haystack）から特定の事実（Needle）を見つけ出す「Needle-In-A-Haystack (NIAH)」[1] は，合成データであるため，長さを自由に調整できる．しかし，単一の事実を検索する初期の NIAH は，最新のモデルにとっては容易なタスクとなりつつある．そのため，評価の焦点は単純な検索から，より複雑な推論や情報の統合へと移行している．LangChain は複数の事実を検索・統合す

る「Multi-needle」タスク [5] を提案し，Yu らは時系列順や論理順で再構成する Sequential-NIAH ベンチマーク [6] を提案した．

3 ベンチマークの構築

本研究では，日本語人事労務ドメインに特化した NIAH ベンチマークを構築するため，架空企業「株式会社 N.I.A. ヘイスタック」を設定し，その企業の人事労務データを Haystack（干し草の山）として構成した．Haystack は，全社規定（就業規則，給与テーブル，育児・介護休業規定，ハラスメント防止規定）と従業員別データ（従業員 60 名分の勤怠ログ，雇用契約書履歴，1on1 面談記録）から構成される．従業員 1 人あたりのデータ量は約 10k 文字となるよう調整した．プライバシー保護と法務上の制約を考慮し，実在する企業や個人のデータではなく，合成データとして構築した．この合成データの作成の補助には，AI エディタである Cursor（Agent Mode）を活用した．具体的には，Gemini シリーズおよび GPT シリーズのモデルを用いて，生成された結果の目視確認と修正依頼を対話的に繰り返しながら作成した．最終的には目視で確認し，必要に応じて手動修正も加えた．本ベンチマークでは，単一データ検索（Single-Needle），複数データ検索（Multi-Needle），時系列データ検索（Sequential-Needle）の 3 種類のタスクタイプを設計した．具体的例を表 1 に示す．

本実験では，従業員 1 名に対しこれら 3 つのタスクタイプをそれぞれ 3 件ずつ，計 9 つのタスクを割り当てている．全従業員 60 名分を合計すると，各タスクタイプ 180 件ずつ，総計 540 件のタスクから構成されるデータセットとなる．

評価は，全タスクに対する Gold Standard（正解データ）を JSONL 形式で準備し，Exact Match（完全一致）による自動評価を実施する．構造化出力が必要なタスクについては，Pydantic モデルによる回答形式の検証を実施する．評価の一貫性を保つため，ドキュメントの結合順序は固定されている．

4 評価実験

4.1 評価対象モデル

本ベンチマークでは，以下の LLM を評価対象とした．

- GPT 系 (GPT-5, GPT-5-mini)
- Gemini 系 (gemini-3-pro-preview, gemini-3-flash-preview) ※ thinking level は high に設定
- Claude 系 (claude-opus-4-5@20251101, claude-sonnet-4-5@20250929)

これらはクローズドモデルであるが，以下の理由により選定した．

- 既に産業界では広く使われており，検証結果の実応用上の貢献度が高いこと．
- LLM がブラックボックスであってもベンチマークの妥当性は検証可能であること．
- ユーザー評価のリーダーボード [7] においてオープンソースのモデルよりも良い結果を記録していること．

4.2 実験設定およびパラメータ

本実験では，コンテキスト長が性能に与える影響を評価するため，従業員 ID を段階的に増加させることでコンテキスト長を変化させた．具体的には文字数ベースで最小 44k から最大はモデル毎に GPT 系は 361k，Gemini 系は 535k，Claude 系 184k とした．これは，GPT 系と Claude 系が左記のコンテキスト長で最大値付近となるためである．

5 実験結果と考察

5.1 タスク種別による性能評価

図 1 にコンテキスト長 184k（15 従業員，計 135 質問）におけるタスクタイプ別の精度比較結果を示す．この結果から，Single-Needle，Multi-Needle，Sequential-Needle の順に精度が低くなっている傾向が確認できる．しかし，gpt-5-mini のみ Single-Needle が最低の精度となった．gpt-5-mini の間違いは，他のモデルに比べ指示や体裁を守らない傾向であることが確認された．

5.2 コンテキスト長に対するスケーラビリティとモデル間比較

図 2 に Sequential-Needle タスクのモデル間の性能比較結果を示す．この結果から，コンテキスト長を増やすと精度が劣化すること，同系統の下位モデルは上位モデルに精度が劣ることが示唆される．しかし，gemini-3-flash において上位モデルとの精度の逆転現象が起きている．これは他のベンチマークでも同様の逆転が報告されており [8]，蒸留等で作成した単純な下位モデルでないことが示唆される．

表 1: 各タスクの具体例

タスクタイプ	具体例
Single-Needle	従業員 E001 の 2024 年 06 月における時間外労働の合計時間数を教えてください。
Multi-Needle	従業員 E001 の最新の契約に基づく等級は？
Sequential-Needle	従業員 E001 の役職の履歴を、日付と役職が分かる時系列の JSON 形式で答えてください。

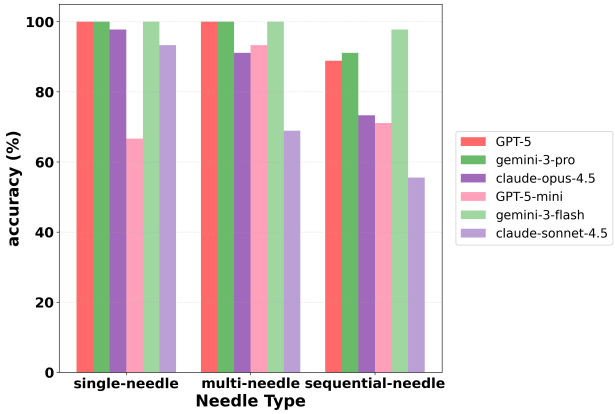

図 1: コンテキスト長 184k のタスク別精度比較

図 2: Sequential-Needle タスクの性能比較

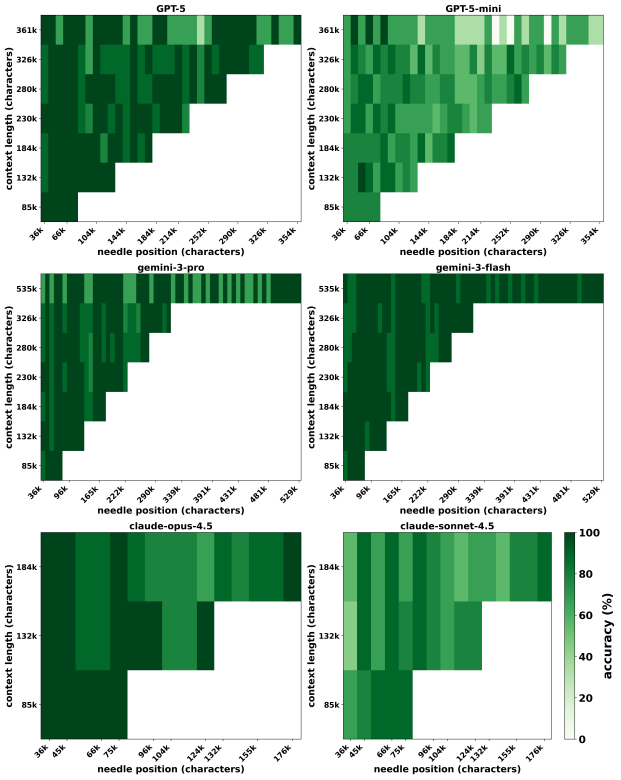

図 3: Needle 位置と精度の関係

5.3 Needle 位置が精度に与える影響の検証

図 3 は縦軸をコンテキスト長，横軸をニードル位置，色を正解率としたヒートマップである．この図から，GPT，Claude 系においてはコンテキストの中央部分での精度が低下する「Lost in the Middle」現象 [10] が現れていることが分かる．特に，コンテキスト長が増加するにつれて，中央部分での精度低下が顕著に観察された．

5.4 エラー分析とベンチマークの妥当性考察

評価において不正解となった失敗事例を Cursor の Agent モードを補助に目視分析した結果，失敗原因は「ベンチマークの改善余地があるもの」と「モデルの能力に起因するもの」に大別された．表 2 にベンチマークの改善余地がある失敗例を示す．特に「役職履歴における部署情報の混入」や「時系列データにおける余剰エントリの追加」は，ベンチマークの記述が複数の解釈を許容するために発生している．

一方で，ベンチマーク側は適切に機能しており，モデルの純粋な能力不足によって発生した失敗例とその内訳を図 4 に示す．主な要因は，フォーマット遵守能力，数値計算・集計精度，日付抽出能力の 3 点である．

6 結論

本研究では，日本語の人事労務ドメインに特化した独自の NIAH ベンチマークを構築し，最新のロングコンテキスト LLM の性能を多角的に評価した．実験の結果，Single や Multi より，Sequential-Needle

表 2: ベンチマークの改善余地がある失敗例とそれに対する具体的な改善策

失敗パターン	発生件数	エラーの態様	ベンチマーク改善方針
役職履歴における部署情報の混入	79 件	**正解**: `[{"role": "部長"}]` **出 力**: `[{"role": "開 発 本 部 部長"}]`	質問文へ「部署名は含めず，役職名のみを回答すること」という制約を明記する．
時系列データにおける余剰エントリの追加	102 件	**正解**: 採用時と昇進時の 2 点のみ **出力**: 役職変更のない契約更新時を含む 4 点 ※履歴の網羅性に関する解釈の不一致	質問文に「役職に変更があったタイミングのみを記録し，維持・更新時は含めない」と抽出条件を厳密化する．

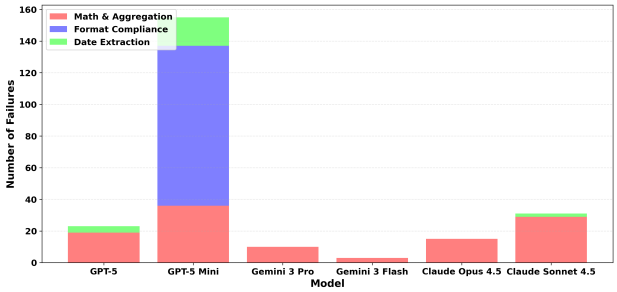

図 4: モデル別の失敗件数とその内訳

においてモデル間の性能差が顕著になることが明らかとなった．また，日本語の実務ドメインにおいても，コンテキスト中央部で精度低下する「Lost in the Middle」現象が確認され，長大な実務ドキュメントを扱う際のリスクが示された．

7 今後の展望

本研究で得られた知見を基に，今後は次の 3 点に注力する．第一に，失敗事例の分析で判明したプロンプトの曖昧さを解消し，数百万トークン級への Haystack の拡張や非定型なコミュニケーションログの追加により，さらに実務に近い複雑なシナリオでの検証を目指す．第二に，SummHay タスク [11] のような高難易度な統合タスクの追加により，より高度な要約・推論タスクを評価する．第三に，RAG システムとの比較評価により，精度・コスト・レイテンシにおけるトレードオフを定量的に評価する．

参考文献

[1] Greg Kamradt. LLMTest_NeedleInAHaystack. GitHub, 2023. https://github.com/gkamradt/LLMTest_NeedleInAHaystack.

[2] Chenxin An, Shansan Gong, Ming Zhong, Xingjian Zhao, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. L-Eval: Instituting Standardized Evaluation for Long Context Language Models. Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 14212-14232, 2024.

[3] Uri Shaham, Maor Ivgi, Avia Efrat, Jonathan Berant, and Omer Levy. ZeroSCROLLS: A Zero-Shot Benchmark for Long Text Understanding. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 7977–7989, 2023.

[4] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1845–1862, 2024.

[5] LangChain. Multi-Needle in a Haystack, 2024. https://blog.langchain.com/multi-needle-in-a-haystack/.

[6] Yifei Yu, Qian-Wen Zhang, Lingfeng Qiao, Di Yin, Fang Li, Jie Wang, Zengxi Chen, Suncong Zheng, Xiaolong Liang, and Xing Sun. Sequential-NIAH: A Needle-In-A-Haystack Benchmark for Extracting Sequential Needles from Long Contexts. Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP 2025), pp. 16538–16555, 2025.

[7] LMSYS Org. LMSYS Chatbot Arena Leaderboard, (2025/12/25 閲覧). https://lmarena.ai/ja/leaderboard.

[8] Tulsee Doshi. Gemini 3 Flash：スピードを追求した最先端知能，2025. https://blog.google/intl/ja-jp/company-news/technology/gemini3-flash/.

[9] Carlos E. Jimenez, John Yang, Alexander Wettig, et al. SWE-bench: Can Language Models Resolve Real-World GitHub Issues?. International Conference on Learning Representations (ICLR), 2024.

[10] Nelson F. Liu, Kevin Lin, Michele Bevilacqua, Fabio Petroni, Ashwin Paranjape, and Percy Liang. Lost in the Middle: How Language Models Use Long Contexts. Transactions of the Association for Computational Linguistics, Vol. 12, pp. 157-173, 2024.

[11] Philippe Laban, Alexander R. Fabbri, Caiming Xiong, Chien-Sheng Wu. Summary of a Haystack: A Challenge to Long-Context LLMs and RAG Systems. Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP 2024), 2024.