タスク指示プロンプトによる大規模言語モデル内部のユーザー表現変化と汎用性の分析

日高逸稀 [1]　小町守 [1]　欅惇志 [1]
[1] 一橋大学大学院
{dm250012@g, mamoru.komachi@r, a.keyaki@r}.hit-u.ac.jp

概要

ユーザー情報を埋め込み表現に変換するユーザー埋め込み手法の多くは特定の下流タスクに最適化され，他タスクへの転用が難しい．本研究では，同一のユーザー行動履歴に対してタスク指示プロンプトを変更することで，LLM の内部表現からタスク指示特化のユーザー埋め込みを動的に得られるかを検証する．複数の下流タスクにおける評価結果から，本提案手法が追加学習を伴わず軽量かつ高い汎用性を有することを確認した．

1　はじめに

ユーザー埋め込みとは，ユーザーが過去に行った一連の行動（映画の視聴，商品の購入など）を時系列に記録したユーザー行動履歴（以降，ユーザー履歴）に基づき，ユーザー間の類似性や関係性をベクトル空間上で扱えるようにした表現である [1]．近年，ユーザー埋め込みの有用性は推薦性能の向上にとどまらず，ユーザークラスタリングやパーソナライゼーションの高度化など，ユーザー理解を要する幅広いタスクにおいて高まっている [2]．

しかし，既存のユーザー埋め込み手法の多くは，特定の下流タスクに最適化されており，他のタスクへの転用が困難であるという汎用性の課題がある [3]．一方，マルチタスク学習に基づく汎用的な手法 [4] も提案されているものの，タスクによらず単一のベクトルでユーザーを表現するため，各タスクに対する最適性が十分でない場合が多い．このようにユーザー埋め込み手法における課題は，多様な下流タスクに対応可能な汎用性を確保しきれていない点にある．さらに実運用の観点では，複雑なモデル構造や追加学習を要しない簡便な手法が望まれ，個別タスクにおいても十分な性能が期待される．

そこで本研究では，LLM の内部表現を埋め込みとして利用する先行研究 [5, 6] に着想を得て，同一のユーザー履歴に対してタスク指示プロンプトを変更することで，LLM の内部表現が下流タスクに応じたユーザー埋め込みとして動的に制御可能かを検証する．LLM によってタスク特化した埋め込みを得られれば，単一モデルによる柔軟かつ汎用的なユーザーモデリングが可能となる．

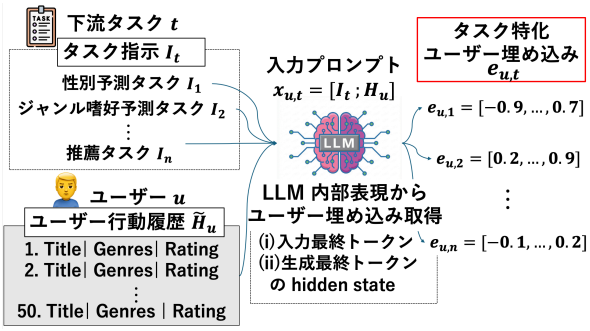


図 1　本研究の概要図

具体的には，図 1 のようにタスク指示（例：Predict the gender based on user history.）とユーザー履歴をプロンプトとして与え，ユーザー埋め込みとして用いる hidden state の抽出位置を (i) 入力最終トークン，(ii) 生成最終トークンの 2 方式から比較し，それらが下流タスク性能および汎用性に与える影響を，複数の下流タスクに対する評価を通じて分析した．実験結果から，タスク指示プロンプトのみを変更することで，同一ユーザー履歴からタスク特化したユーザー埋め込みを動的に生成可能であることを確認し，特に生成最終トークンの内部表現が汎用的なユーザー埋め込み手法となりうることが示された．

2　関連研究

ユーザー埋め込み手法　従来のユーザー埋め込み手法は，下流タスク性能と汎用性の両立に課題があり，同一モデルで多様な下流タスクに特化したユーザー埋め込みを獲得可能な手法が求められている．

ユーザー埋め込みの研究は，主に情報推薦分野において発展してきた．近年では，RNN を用いる GRU4Rec [7] や，Transformer ベースの SASRec [8]，BERT4Rec [9] などが提案されており，推薦タスクにおいて高精度なユーザー埋め込みの獲得を実現している．一方で，これらは単一タスクの性能最大化を目的として学習されるため，推薦タスク以外の下流タスクへの転用が困難であるという課題が残る．

この問題に対し，行列分解 (MF) に基づく協調フィルタリング [10] や，マルチタスク学習に基づくユーザー埋め込み手法 [4] など，汎用的なユーザー埋め込みを獲得する手法も提案されてきた．しかし，いずれもユーザーを固定のベクトルとして表現するため，同一のユーザー埋め込みを複数タスク間で再利用可能であるものの，各タスクに対する最適性が失われるという課題が指摘されている [11]．

近年では，ユーザー履歴を圧縮する専用の User Encoder と LLM を組み合わせた手法や [12, 13]，LLM 自体を推薦タスク向けに適応させる手法も提案されている [14, 15]．これらは性能と汎用性の両立を志向するものであるが，追加学習やモデル構成の複雑化を伴う点で，下流タスクの切り替えが困難であるという実運用上の課題が残る．本研究は，こうした専用構成を導入せず，プロンプト制御のみにより単一の LLM でタスク特化かつ汎用的なユーザー埋め込みを実現可能か検討する．

LLM の内部表現を用いた埋め込み手法　近年，タスク指示に応じて埋め込み表現を動的に変化させる埋め込み手法が注目されている [16]．中でも，InBedder [6] は，LLM にタスク指示と埋め込み対象テキストを同時に与え，生成トークンに対応する内部表現を埋め込みとして利用することで，タスク特化した埋め込みを得る手法として有効性を示している．一方でこれらは，単一の短文テキストを主な対象としており，長期にわたるユーザー履歴のような長大かつ構造的なシーケンスデータを入力とした埋め込み獲得については，十分に検証されていない．

3　分析手法

本研究の目的は，同一のユーザー履歴に対して，タスク指示プロンプトを与えた時に，内部表現のどの位置をユーザー埋め込みとして用いるかに着目して，下流タスクに適したユーザー埋め込みを単一の LLM から獲得可能か検証することである．本節では，LLM を用いたユーザー埋め込みの枠組みを定式化し，埋め込み抽出位置が異なる 2 種類のユーザー埋め込み手法を比較する．

3.1　問題設定

ユーザー集合を $\mathcal{U}$，アイテム集合を $\mathcal{I}$ とする．ユーザー $u \in \mathcal{U}$ の行動履歴系列を

$$H_u = [i_1, i_2, \ldots, i_L], \quad i_\ell \in \mathcal{I} \tag{1}$$

と表す．各アイテム i に付与されているタイトルやジャンルなどのメタ情報をテキストとして連結し，ユーザー履歴 H_u をテキスト系列 $\tilde{H}_u$ に変換する．

本研究では，ユーザー埋め込みを用いる下流タスクの集合を $\mathcal{T}$ とし，各タスク $t \in \mathcal{T}$ に対して自然言語によるタスク指示 I_t を設定する．タスク指示とユーザー履歴を結合して構成する history-and-task プロンプト $x_{u,t}$ は下記の通り定義される．

$$x_{u,t} = [I_t; \tilde{H}_u] \tag{2}$$

本研究では，LLM に対してプロンプト $x_{u,t}$ を入力した際に得られる hidden state をユーザー埋め込み $\mathbf{e}_{\mathbf{u},\mathbf{t}} \in \mathbb{R}^d$ として定義し，下流タスク t に対応する予測器 h_t を介してラベル $y_{u,t}$ を予測する枠組みを構築し，性能を比較する．なお，付録 B にてタスク指示プロンプトを用いない場合の検証を行う．

3.2　ユーザー埋め込みの抽出方法

本研究では，LLM の hidden state をユーザー埋め込みとして用いる．hidden state の抽出位置の違いにより，以下の 2 手法を比較する．

Prompt-last ユーザー埋め込み　プロンプト $x_{u,t}$ の最終トークンに対応する hidden state をユーザー埋め込み $\mathbf{e}_{\mathbf{u},\mathbf{t}}^{\text{prompt}}$ として抽出する．

Gen-last ユーザー埋め込み　プロンプト $x_{u,t}$ に対して生成された出力の最終トークンの hidden state をユーザー埋め込み $\mathbf{e}_{\mathbf{u},\mathbf{t}}^{\text{gen}}$ として抽出する．

4　実験

本節では，3.1 節で定義した問題設定において，3.2 節で提案した 2 種類の LLM-based ユーザー埋め込み手法の有効性を評価する．

4.1　実験設定

データセット　本研究では，ユーザーの映画視聴履歴と属性ラベルを含む MovieLens-1M データセット [17] を用いる．表 1 に，データセット統計を示す．各映画にはタイトルに加えて，映画ジャンルお

表 1 MovieLens-1M データセットの基本統計量

統計量	値
ユーザー数	6,040
アイテム数（映画）	3,883
評価数	1,000,209
映画ジャンル数	18
ユーザーあたり評価数（平均）	165.6
ユーザーあたり評価数（中央値）	96
ユーザーあたり評価数（最小 / 最大）	20 / 2,314

よび 1–5 の評価値，各ユーザーについて年代，性別，職業の属性データも付与されている．

検証タスクと評価指標　提案手法の汎用性を評価するため，性質の異なる以下のタスクを設定する．

- **ユーザー属性予測**
  - **Age**: ユーザーの年齢を 7 つのグループ (Under 18, 18-24, 25-34, 35-44, 45-49, 50-55, 56+) から予測する．
  - **Gender**: 性別 (Male / Female) を予測する．
  - **Occupation**: 21 種類の職業カテゴリ (programmer, student, retired 等) から予測する．
- **ジャンル嗜好予測**
  - **Favorite Genre**: 全視聴履歴の中で，評価 4 以上を与えた映画のジャンルにおいて，最も出現頻度が高いジャンルを予測する．
  - **Recent Favorite Genre**: 直近 20 件の視聴履歴の中で，評価 4 以上を与えた映画のジャンルにおいて，最も出現頻度が高いジャンルを予測する．
- **推薦タスク**
  - **Next Item Prediction**: 次に視聴する映画を予測する．[1)]
  - **Next Genre Prediction**: 次に視聴する映画のジャンルを予測する．

評価指標として，Next Item Prediction においては HR@10 および NDCG@10，その他タスクは Accuracy および Macro-F1 を用いる．手法間の全体的なタスク性能を評価するために，NextItem を除く 6 タスクにおける各指標の算術平均も併せて算出する．

比較手法

- **LLM-based (Prompt-last / Gen-last)**: 3.2 節で定義した提案方式を用いる．LLM は，Llama-3.1-8B-Instruct[2)] を採用した．生成時には，貪欲法で max new tokens = 120 を設定した．
- **LoRA**: 提案手法と同じ LLM を用い，プロンプト $x_{u,t}$ の最終トークンの 最終層 hidden state を分類ヘッドに入力し，タスクラベル $y_{u,t}$ を予測するように LoRA[18] で学習する．学習後モデルにおける同 hidden state をユーザー埋め込みとする．本手法は，追加学習を許した場合の LLM 内部表現を用いた高性能手法の参考値として比較し，どの程度この上限に迫れるか，およびどのようなタスクで差が生じるかを分析する．
- **推薦タスク特化型モデル (GRU4Rec / SASRec)**: モデルの最終層におけるユーザー表現を用いる．これらのモデルは構造上の制約に基づき，入力は映画タイトル系列のみとする．
- **MF-BPR**: 行列分解により得られるユーザー潜在ベクトルを用いる．

プロンプト設計　プロンプトは 3.1 節で定義した提案手法の history-and-task を用いる．ユーザー履歴 $\tilde{H}_u$ は全タスクで共通であり，各アイテムを `Title: <title> | Genres: <g1> | <g2> | ... | Rating: <r>/5` の 1 行として時系列順に番号付きで並べた．

タスク指示プロンプト I_t は，System プロンプトで役割付与，User プロンプトでタスク指示および出力形式の明示から成る．タスク指示において生成過程で推論を促すために，タスクを解く上での 3 つの手掛かりを列挙させたうえで回答を出力させるようにプロンプトを設計した．これにユーザー履歴 $\tilde{H}_u$ を結合し，最終的なプロンプト $x_{u,t}$ を構築する．使用したプロンプトの詳細を付録 A に示す．

評価プロトコル　全手法を公平に比較するため，probing 設定を採用する．具体的には，直近 50 件に切り詰めたユーザー履歴を入力として得られたユーザー埋め込みを，ロジスティック回帰を用いて，下流タスク性能を評価する．[3)]LLM-based 手法では，全 32 層の hidden state をユーザー埋め込みとして probing を行い最高性能の層のスコアを報告する．

学習を要する手法は，ユーザー 6,040 件のうち 1,000 件を学習用，残り 5,040 件を評価用ユーザー集合として用いる．評価用ユーザー集合からユーザー埋め込みを抽出し，評価用ユーザーをロジスティック回帰の学習用と評価用に 8:2 で分割する．

1) 予測対象のアイテムは，データセット全体で 200 件以上の評価を受けている映画に対象を限定した．これにより，予測対象のアイテム数は 3,883 件から 1,426 件に絞り込まれる．

2) https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

3) 本設定では，推薦特化モデルも専用出力層を用いず，最終層表現のみを利用する．そのため，Next Item Prediction の性能を過小評価し得るが，本研究の目的はユーザー埋め込みとしての性能比較であるため，同一の probing 設定を適用する．

表 2　各下流タスクにおける probing 評価結果（太字は最良値，下線は次点を表す）

Method	Gender		Age		Occupation		FavGenre		RecFavGenre		NextGenre		NextItem@10		Avg.	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	HR	NDCG	Acc.	F1
Prompt-last	.624	.731	.248	.205	.084	.066	.566	.361	.427	.285	.230	.104	.0205	.0094	.363	.292
Gen-last	.692	.781	.255	.200	.086	.064	.784	.625	.596	.435	.241	.084	.0205	.0097	.442	.365
LoRA	**.714**	**.785**	**.328**	**.282**	**.111**	**.083**	**.878**	**.793**	**.851**	**.804**	**.348**	**.165**	.0466	.0243	**.538**	**.485**
MF-BPR	.706	.671	.313	.281	.075	.072	.713	.580	.559	.447	.146	.089	.0479	.0231	.419	.357
GRU4Rec	.561	.531	.154	.146	.032	.028	.182	.132	.245	.225	.076	.067	.0603	.0292	.208	.188
SASRec	.589	.565	.175	.167	.047	.045	.355	.231	.274	.225	.124	.084	**.0630**	**.0317**	.261	.220

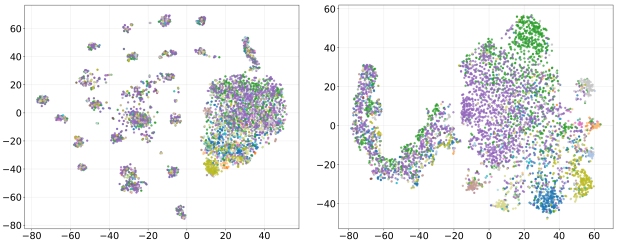
4.2　結果

表 2 に，各下流タスクに対する probing 評価結果を示す．まず全体的な傾向として，Gen-last は LoRA を除くと，ベースラインと比較して平均して最も高い性能を示した．また，Prompt-last と Gen-last を比較すると，Gen-last が多くのタスクにおいて高い性能を示した．これは，追加学習を行わない制約下において，Gen-last が最も高品質で汎用的なユーザー埋め込みを獲得可能であることを示唆している．

Gen-last は，特にジャンル嗜好予測 (FavGenre, RecFavGenre) で LoRA との性能差が相対的に小さく，高性能だった．これは，履歴から正解ラベルが一意に定まるタスクでは，埋め込むべき情報が明確であるため，生成過程を通じてタスク指示に関連する履歴情報が取捨選択され，内部表現に強く集約されるためと考えられる．具体的には，Gen-last では LLM がタスク指示に従い回答を生成し推論する中で，履歴全体に分散していた情報からジャンルや評価といったタスクに関連する情報が集約され，最終トークンの内部表現に統合される．その結果，タスク指示に特化したユーザー埋め込みが動的に得られたと考えられる．この結果は，低コストな方法で微調整を代替可能であることを示唆している．

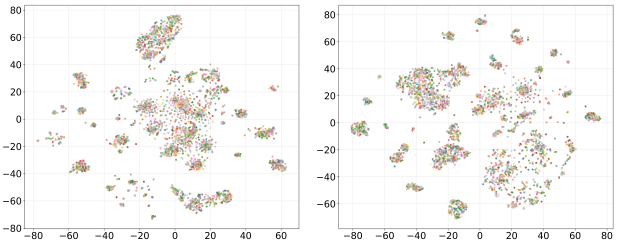
一方で，Age や Next Item Prediction のように，履歴のみから正解が一意に定まらないタスクでは，ユーザー履歴中のどの情報を取捨選択すべきかが明確でない．その結果，Gen-last は LoRA に比べて一貫した性能改善には至らなかった．この結果は，これらのタスクにおいては，何を重要特徴とみなすかをモデル内部に固定的に獲得するための追加学習が有効であることを示している．なお，履歴長を変化させた時の検証については，付録 C に示す．

また，これらの結果は埋め込み空間においても整合的であった．FavGenre および Occupation を対象に，最高性能の層におけるユーザー埋め込みを



(a) Prompt-last　　(b) Gen-last

図 2　FavGenre におけるユーザー埋め込みの可視化



(a) Prompt-last　　(b) Gen-last

図 3　Occupation におけるユーザー埋め込みの可視化

t-SNE により 2 次元に可視化した結果を図 2，図 3 に示す．FavGenre では，特に Gen-last において，クラスごとのクラスタ構造が比較的明瞭に観察された一方で，Occupation では，クラス間の分離は限定的であった．これは，前述の履歴から正解ラベルが一意に定まるか否かで傾向が違うという議論と整合的であり，本結果からも生成最終トークンを用いる Gen-last の有効性が確認できた．

5　おわりに

本研究では，LLM の内部表現からタスク指示に特化したユーザー埋め込みを獲得できるかを検証した．生成最終トークンを用いる Gen-last は，平均して高い性能を示し，追加学習を伴わない条件下でもタスク指示に関連する情報が内部表現に強く集約される傾向が確認された．

今後は，別ドメインデータセットにおける有効性の検証や，LLM のモデルサイズを変更した場合の性能変化について，より詳細な分析を進める．

謝辞

本研究の一部は，JSPS 科研費（基盤研究 (B) (課題番号: 23H03686, 25K03178)，基盤研究 (C) (課題番号: 24K15066)，令和７年度次世代人工知能技術等研究開発拠点形成事業「生成 AI モデルの透明性・信頼性の確保に向けた研究開発拠点形成」，株式会社デンソーアイティーラボラトリとの共同研究の支援による．ここに記して謝意を表す．

参考文献

[1] Shiwei Li, Huifeng Guo, Xing Tang, Ruiming Tang, Lu Hou, Ruixuan Li, and Rui Zhang. Embedding compression in recommender systems: A survey. **ACM Computing Surveys**, Vol. 56, No. 5, pp. 130:1–130:21, 2024.

[2] Erasmo Purificato, Ludovico Boratto, and Ernesto William De Luca. User modeling and user profiling: A comprehensive survey. arXiv.2402.09660, 2024.

[3] Chenglin Li, Yuanzhen Xie, Chenyun Yu, Bo Hu, Zang Li, Guoqiang Shu, Xiaohu Qie, and Di Niu. One for all, all for one: Learning and transferring user embeddings for cross-domain recommendation. In **Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM)**, 2023.

[4] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H. Chi. Modeling task relationships in Multi-task learning with Multi-gate mixture-of-experts. In **Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)**, 2018.

[5] Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. Scaling sentence embeddings with large language models. In **Findings of the Association for Computational Linguistics: EMNLP**, 2024.

[6] Letian Peng, Yuwei Zhang, Zilong Wang, Jayanth Srinivasa, Gaowen Liu, Zihan Wang, and Jingbo Shang. Answer is all you need: Instruction-following text embedding via answering the question. In **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)**, 2024.

[7] Bal'azs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In **Proceedings of the 4th International Conference on Learning Representations (ICLR)**, 2016.

[8] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In **Proceedings of the IEEE International Conference on Data Mining (ICDM)**, pp. 197–206, 2018.

[9] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. BERT4rec: Sequential recommendation with bidirectional encoder representations from Transformer. In **Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)**, pp. 1441–1450, 2019.

[10] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. **Computer**, Vol. 42, No. 8, pp. 30–37, 2009.

[11] Liangcai Su, Junwei Pan, Ximei Wang, Xi Xiao, Shijie Quan, Xihua Chen, and Jie Jiang. STEM: Unleashing the power of embeddings for Multi-Task recommendation. In **Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI)**, 2024.

[12] Lin Ning, Luyang Liu, Jiaxing Wu, Neo Wu, Devora Berlowitz, Sushant Prakash, Bradley Green, Shawn O'Banion, and Jun Xie. User-LLM: Efficient LLM contextualization with user embeddings. In **Companion Proceedings of the ACM on Web Conference 2025 (WWW '25)**, 2025.

[13] Sumanth Doddapaneni, Krishna Sayana, Ambarish Jash, Sukhdeep Sodhi, and Dima Kuzmin. User embedding model for personalized language prompting. In **Proceedings of the 1st Workshop on Personalization of Generative AI Systems (PERSONALIZE 2024)**, 2024.

[14] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. TALLRec: An effective and efficient tuning framework to align large language model with recommendation. In **Proceedings of the Seventeenth ACM Conference on Recommender Systems (RecSys)**, 2023.

[15] Yingzhi He, Xiaohao Liu, An Zhang, Yunshan Ma, and Tat-Seng Chua. LLM2rec: Large language models are powerful embedding models for sequential recommendation. In **Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)**, 2025.

[16] Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. One embedder, any task: Instruction-Finetuned text embeddings. In **Findings of the Association for Computational Linguistics: ACL 2023**. Association for Computational Linguistics, 2023.

[17] F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. **ACM Transactions on Interactive Intelligent Systems**, Vol. 5, No. 4, pp. 19:1–19:19, 2015.

[18] Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In **Proceedings of the International Conference on Learning Representations (ICLR)**, 2022.

表 3　history-only の probing 評価結果（太字は最良値，下線は次点を表す）

Method	Gender		Age		Occupation		FavGenre		RecFavGenre		NextGenre		NextItem@10		Avg.	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	HR	NDCG	Acc.	F1
Prompt-last (history)	.663	.750	**.260**	**.225**	.080	.065	.522	.301	.392	.269	.223	.090	**.0274**	**.0153**	.357	.283
Gen-last (history)	.655	.750	.246	.196	**.089**	.063	.552	.254	.350	.121	**.245**	.099	.0205	.0099	.356	.247
Prompt-last (hist + task)	.624	.731	.248	.205	.084	**.066**	.566	.361	.427	.285	.230	**.104**	.0205	.0094	.363	.292
Gen-last (hist + task)	**.692**	**.781**	.255	.200	.086	.064	**.784**	**.625**	**.596**	**.435**	.241	.084	.0205	.0097	**.442**	**.365**

A　プロンプト例

Next Item Prediction にて使用したプロンプト例.

```
System:
You are an expert movie recommendation analyst.
Provide answers based on the user's history.
User:
Predict the next movie this user is likely to watch
based only on the viewing history below. Based on the
viewing history, predict the next movie this user is
likely to watch. First, identify three cues that support
your inference, and then state the final movie title.
All cues must be grounded in the viewing history
(e.g., sequel likelihood, director affinity, recent
genre trend, or engagement with specific franchises).
Compare possibilities and choose the movie that is most
strongly supported.
You must strictly follow the output format below:
Signals: <cue1>; <cue2>; <cue3>
Next Movie: <Title>
User history (most recent last):
1. Title: Pulp Fiction (1994) |
   Genres: Crime | Drama |
   Rating: 5/5
2. Title: Heavenly Creatures (1994) |
   Genres: Drama | Fantasy | Romance | Thriller |
   Rating: 4/5
3. Title: Beauty and the Beast (1991) |
   Genres: Animation | Children's | Musical |
   Rating: 4/5
```

B　history-only の結果

タスク指示をプロンプトとして明示的に与えることの有用性を検証するために，ユーザー履歴のみからなる history-only プロンプトの実験結果を表 3 に示す．この結果から，タスク指示をプロンプトとして明示的に与えた Gen-last が平均して最も高い性能を示しており，提案手法の有用性が確認できた．

C　履歴長の影響

LLM-based ユーザー埋め込みが長い履歴系列に対して有効であるかを検証するため，ユーザー属性予測タスクを対象として，入力履歴長を $\{10, 20, 30, 40, 50\}$ と変化させて実験を行った．本実験の目的は，既存研究では十分に検討されていない長いユーザー履歴を入力とした場合に，履歴長が長くなることで LLM の内部表現を用いたユーザー埋め込みにどのような影響があるのかを検証する点にある．履歴長 10 を基準とした相対的な性能の結果を図 4 に示す．

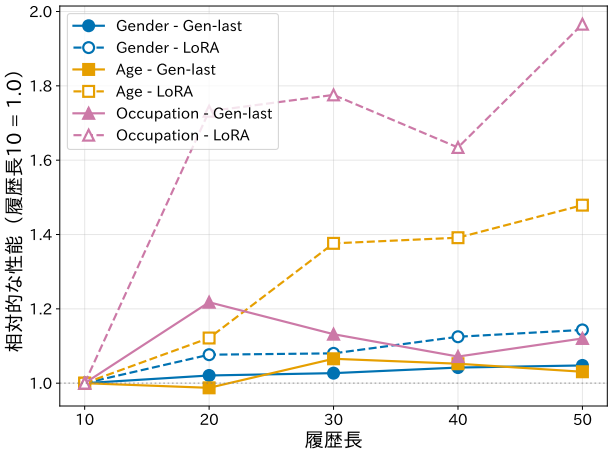


図 4　履歴長を変化させた時の probing 性能

結果より，Gen-last および LoRA のいずれにおいても，履歴長の増加に伴い性能が向上する傾向が確認された．この結果は，履歴長の増大により観測可能なユーザー履歴が増えることで，当該タスクの識別が容易になるという難易度変化の観点と整合的であり，LLM の内部表現を用いた埋め込み手法が，ユーザー履歴という長大なシーケンスデータに対しても，タスクに応じた有用な特徴を保持できることが示唆された．