

逆プロンプトを用いたコールドスタート推薦

草野 元紀¹

¹ 日本電気株式会社
g-kusano@nec.com

概要

本研究では、新商品などの教師データに該当商品がなくて予測が困難になるコールドスタート推薦に取り組む。この問題に取り組むために、従来は商品名やカテゴリーといった補助情報を用いる推薦システムが提案されていたが、十分な量の訓練データが必要であった。近年では、大規模言語モデル (LLM) を用いることで教師データなしにコールドスタート問題を解けるようになったが、運用時のコスト面に課題があった。本論文では、LLM をデータ拡張機として活用し、教師データ収集とコスト効率の両課題を解消する RevAug を提案する。RevAug のアイデアは、ユーザがその商品を好きかどうかを予測させる従来の推薦プロンプトを、ユーザが好きなような商品を生成させるプロンプトに変換し、それにより得られる出力を擬似サンプルとして学習データに活用したことである。4つの実データを用いた数値実験では、RevAug は少ない教師データで高い推薦精度を達成し、LLM の処理時間と利用料金を大幅に削減した。

1 序論

推薦システムはECサイトや広告配信などで広く利用されており、そこで扱う問題はユーザがアイテムをどのくらい好むかを予測することである。古典的には行列分解に代表される協調フィルタリング [1, 2, 3] が、教師データに含まれるユーザ、アイテム、反応の3つ組からその対応関係を学習し、ユーザがまだ見ていないアイテムへの反応を予測させている。新商品などの教師データにないアイテムを推薦する際、古典的な行列分解では対応するアイテム表現を求められないため予測ができず、これはコールドスタート問題として知られている難問である。

この問題を解決するため、アイテムに付随する商品名やカテゴリーといった補助情報を利用した推薦システム [4, 5, 6, 7, 8] が提案されている。これらの手

法は、教師データ内のユーザやアイテムの協調埋め込みに加え、補助情報から得られるテキスト表現も学習に活用することで、コールドスタート問題に取り組んでいる。しかし、高精度な推薦を行うためには十分な量と質のデータが必要という課題がある。

近年、大規模言語モデル (Large Language Model, LLM) の技術進歩が著しく、教師データがなくても高精度な推薦が可能であることが知られており、それゆえコールドスタート推薦にも応用されている [9, 10, 11, 12, 13, 14, 15, 16]。しかし、これらの方法は推論時のコストが大きいという課題がある。例えば、1000人のユーザと100個のアイテムの反応を予測する場合、LLMによる推論処理が10万回必要となり、膨大な処理時間と利用料金がかかる。

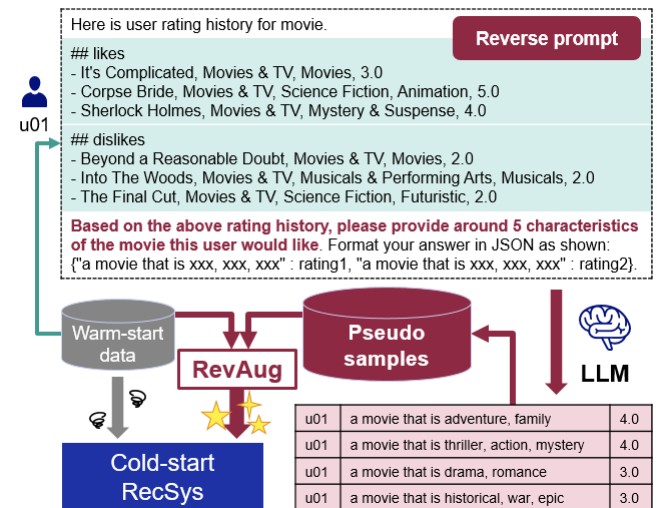


図1 提案手法である RevAug の概要。従来のコールドスタート推薦手法はウォームスタートデータのみで訓練されるが、RevAug はこのウォームスタートデータからプロンプトを作成し、LLM を用いて擬似サンプルを生成し、それらを用いて推薦システムを強化する。

本論文では、コールドスタート問題において、推薦精度と推論コストを両立する手法である RevAug を提案する。RevAug のアーキテクチャは、補助情報を利用したコスト効率の高いコールドスタート

推薦手法である Heater [5] と CLCRec [6] を一般化したものである。教師データが限られていることによる推薦性能の低下問題に対しては、LLM を用いたデータ拡張を活用することで対処する。

従来の LLM ベースの推薦システム [12, 14, 15, 17] は、LLM に “{user} は {item} を好むか?” といった質問（プロンプト）を投げかけるアプローチをとっている。これらの研究は、LLM が推薦を高精度に解けるほどまで必要な十分な知識を持っていることを示唆している。本研究では、この知識を引き出すために従来のプロンプトを逆転させ、“{user} はどのようなアイテムを好むか?” という形式に変更した。この逆プロンプトにより、LLM はユーザが好む可能性のあるアイテムに関する補助情報を生成し、それを推薦モデルの訓練に使用する。この出力は擬似サンプルであるため、Heater や CLCRec の基本アーキテクチャに特定の正則化項を加えることで、推薦性能の向上を図った。

RevAug の有効性を評価するために、実データである Amazon Review Dataset を用いた数値実験を行った。その結果、教師データが少ない状況において、RevAug は他の比較手法よりも高い推薦精度を達成した。さらに、RevAug を用いた推論は、LLM ベースの推薦システムと比較して最大で 99.9 % の処理時間を削減した。これらの結果は、提案手法の優位性を実証するものである。

本研究の内容は、国際学会 ACM Recommender Systems (RecSys) 2024 に採択された [18] に基づく。

2 コールドスタート推薦

ユーザ集合を U 、アイテム集合を I 、ユーザがアイテムに反応した際のログを $\mathcal{R} \subset U \times I$ とする。ユーザ u がアイテム i に対して与えた反応は、 $r_{u,i} \in \mathbb{R}$ と表すことにする。この反応は、5 段階評価のような明示的なもの、もしくは、アイテムが購入されたか否かの 2 値 $\{0, 1\}$ の暗黙的なものを含む。

Heater と CLCRec の詳細を説明する前に、いくつかの記法を準備する。アイテムの補助情報は、タイトルやカテゴリなどのテキストとして表現され、SimCSE [19] のような文埋め込み手法で、 $\mathbf{x}_i \in \mathbb{R}^\ell$ としてベクトル化されるとする。 $\mathbf{z}_u, \mathbf{z}_i \in \mathbb{R}^k$ を、後ほど学習されるユーザ u とアイテム i の協調埋め込みとする。また、アイテムの補助情報の文埋め込み \mathbf{x}_i が $\mathbf{W}\mathbf{x}_i \approx \mathbf{z}_i$ のように近づくパラメータ $\mathbf{W} \in \mathbb{R}^{k \times \ell}$ を用意する。他のパラメータ行列とし

て $\mathbf{V}, \mathbf{V}' \in \mathbb{R}^{d \times k}$ を導入し、潜在表現を $\mathbf{f}_i := \mathbf{V}\mathbf{W}\mathbf{x}_i$ 、 $\mathbf{f}_u := \mathbf{V}'\mathbf{z}_u$ と定義する。ユーザ u が見ていないアイテムをランダムに N 個サンプリングしたものを i_1, \dots, i_N とする。また、アイテム $i \in I$ に対し、 i'_1, \dots, i'_N を i とは異なるアイテムとしてランダムにサンプリングする。この時、 $\mathbf{a}, \mathbf{b}, \mathbf{b}_1, \dots, \mathbf{b}_N \in \mathbb{R}^d$ 、 $\tau > 0$ に対して、次のように関数 h を定義する。

$$h(\mathbf{a}, \mathbf{b}, \{\mathbf{b}_n\}_{n=1}^N) := -\log \left(\frac{\hat{p}(\mathbf{a}, \mathbf{b})}{\hat{p}(\mathbf{a}, \mathbf{b}) + \sum_{n=1}^N \hat{p}(\mathbf{a}, \mathbf{b}_n)} \right) \quad (1)$$

ここで、 $\hat{p}(\mathbf{a}, \mathbf{b}) := \exp(\tau^{-1} \langle \mathbf{a}, \mathbf{b} \rangle / \|\mathbf{a}\| \|\mathbf{b}\|)$ とする。関数 h は対照学習（contrastive learning）[20, 21] に由来し、 \mathbf{a} と \mathbf{b} が類似している場合や \mathbf{a} がどの \mathbf{b}_n にも類似していない場合に h は小さくなる。

表 1 誤差関数

	Heater [5] (Explicit)	CLCRec [6] (Implicit)
$\ell_I(u, i)$	$ r_{u,i} - \mathbf{f}_u^T \mathbf{f}_i ^2$	$h(\mathbf{f}_u, \mathbf{f}_i, \{\mathbf{f}_{i_k}\}_{k=1}^N)$
$\ell_A(i)$	$\ \mathbf{W}\mathbf{x}_i - \mathbf{z}_i\ ^2$	$h(\mathbf{V}\mathbf{z}_i, \mathbf{f}_i, \{\mathbf{f}_{i'_n}\}_{n=1}^N)$

これらの記号と表 1 を用いて、Heater と CLCRec の損失関数は、協調埋め込みの誤差 ℓ_I と、補助情報に関する誤差 ℓ_A の和として表現される¹⁾。

$$\mathcal{L}_{Base} := \mathbb{E}_{(u,i) \in \mathcal{R}} [\ell_I(u, i)] + \alpha \mathbb{E}_{i \in I} [\ell_A(i)] + \lambda \|\Theta\|_F^2 \quad (2)$$

ここで、 $\alpha > 0, \lambda > 0, \Theta = (\mathbf{W}, \mathbf{V}, \mathbf{V}', \{\mathbf{z}_u\}_{u \in U}, \{\mathbf{z}_i\}_{i \in I})$ とする。式 (2) の最適化には学習効率を向上させる randomized training [5] として知られる手法を適用し、確率 $p \in [0, 1]$ で $\mathbf{f}_i = \mathbf{V}\mathbf{W}\mathbf{x}_i$ を $\mathbf{V}\mathbf{z}_i$ に置き換える。

3 RevAug

この章では、提案手法である RevAug を紹介する。

3.1 逆プロンプト

序論で述べたように、RevAug では LLM を用いて擬似データを生成する。ここでは、ユーザの行動履歴を few-shot サンプルとしてプロンプトに含める。プロンプトは図 1 や Appendix の図 2 を参照。LLM が生成した全ユーザの擬似データを \mathcal{R}_{Aug} とする。

3.2 訓練

擬似サンプル $(u, i') \in \mathcal{R}_{Aug}$ に対して、LLM によって生成されたアイテムテキスト i' は文埋め込み

1) Heater では \mathbf{V} を用いた変換が言及されていたが、CLCRec では導入されていない。ただし、 \mathbf{V} を単位行列と設定することで CLCRec を再現することができる。ここでは CLCRec に対しても一般化された設定として \mathbf{V} を適用する。

$\mathbf{x}_{i'}$ に変換できる。これにより潜在表現 $\mathbf{f}_{i'} = \mathbf{VW}\mathbf{x}_{i'}$ が求まり、協調埋め込みの誤差 $\ell_I(u, i')$ を計算できる。これらが擬似サンプルであることを考慮し、重みパラメータ $\beta > 0$ を導入し、RevAug の損失関数を次のように定義する。

$$\mathcal{L}_{\text{RevAug}} := \mathcal{L}_{\text{Base}} + \beta \mathbb{E}_{(u, i') \in \mathcal{R}_{\text{Aug}}} [\ell_I(u, i')]. \quad (3)$$

ここで、ハイパーパラメータ β の設定方法について説明する。まず、ウォームスタートデータ \mathcal{R} を訓練集合 \mathcal{R}_{tr} と検証集合 \mathcal{R}_{val} に分割する。次に、 \mathcal{R}_{tr} 上で式 (3) を最小化することでパラメータ Θ_β を得る。訓練後、 \mathcal{R}_{val} 上で推薦スコア $\text{score}(\mathcal{R}_{val}; \Theta_\beta)$ を計算する。このスコアは、明示的フィードバックの場合には Mean Absolute Error (MAE) を、暗黙的フィードバックの場合には normalized Documented Cumulative Gain (nDCG) を用いる²⁾。最後に、選択されたハイパーパラメータとして $\hat{\beta} = \text{argmin}_\beta \text{score}(\mathcal{R}_{val}; \Theta_\beta)$ を採用する。

4 数値実験

RevAug の有効性を検証する数値実験を行った。

4.1 実験設定

4.1.1 データセット

Amazon Review Dataset [22] から、Movie, Music, Book, Grocery の4つを使用した。このデータセットにはユーザの評価 (1 から 5 まで) が含まれており、これを明示的フィードバックとして利用した。暗黙的フィードバックの場合、ユーザの評価が付いているアイテムはすべて正例 ($r_{u,i} = 1$) として扱った。アイテムの補助情報としてタイトルとカテゴリがあり、アイテムテキストを “An item entitled with {title} has {categories} tags.” と設定した。

各データにおいて 100 個のアイテムを選択し、これらをアイテム集合 I として扱った。これらのアイテムに反応したユーザ全員を U として固定した。Movie データに関してはユーザ数が多いため、3000 人に制限し、この人数に達するまでランダムに選択した。 U 内のユーザは I 以外のアイテムにも反応しており、これらから 100 個のアイテムをランダムに抽出し、コールドスタートアイテムとした³⁾。

2) MAE は $\text{score}(\mathcal{R}) := \mathbb{E}_{(u, i) \in \mathcal{R}} [|r_{u,i} - \mathbf{f}_u^T \mathbf{f}_i|]$ によって定義される。nDCG@ K は、全アイテム I のうち、ユーザの好み度合い $\hat{p}(\mathbf{f}_u, \mathbf{f}_i)$ が上位 K に含まれるアイテムから計算される。
3) 各データセットにおける統計量は Appendix の表 5 に示す。

4.1.2 比較手法

RevAug は以下の手法と比較した。

Baseline: コールドスタート推薦では予測がそもそも困難ということから、明示的フィードバックの場合は教師データ内の評価値の平均を出力することが、暗黙的フィードバックの場合はランダムにアイテムを並び替えたものを出力させることが、最低限のベースラインとして機能する。

Base model: 明示的フィードバックには Heater を、暗黙的フィードバックには CLCRec を使用する。これらはデータ拡張なしの RevAug であり、式 (3) における $\beta \equiv 0$ に相当する。

Ablation: LLM によって生成されたデータの有効性を評価するため、意図的に不正確な拡張データ $\mathcal{R}_{\text{Aug}}^{\text{Ab}}$ を作成し、それを学習に使用する。明示的フィードバックの場合、 $\mathcal{R}_{\text{Aug}}^{\text{Ab}} := \{(u, i', 5-r) \mid (u, i', r) \in \mathcal{R}_{\text{Aug}}\}$ とし、暗黙的フィードバックの場合は LLM が好まれないとして出力したアイテムを好まれるアイテムとして使用する (Appendix の図 3)。

ChatGPT: 明示的フィードバックの予測には [15] で説明されているプロンプト (Appendix の図 4) で推論する。コストの都合で、暗黙的フィードバックの予測は行わなかった。

RevAug のパラメータは以下のように設定した: $k = 100$, $d = 100$, $N = 5$, $\tau = 0.1$, $\alpha = 10^{-3}$, $\lambda = 10^{-3}$, $p = 0.5$, $m = 5$ 。補助情報の文埋め込み表現には SimCSE-RoBERTa-large [19] を使用した。RevAug は PyTorch [23] を用いて RAdam [24] により最適化し実装した。これらのパラメータは、RevAug のベースモデルである Heater と CLCRec にも適用した。ハイパーパラメータ調整では、 \mathcal{R} を $|\mathcal{R}_{tr}| : |\mathcal{R}_{val}| = 0.8 : 0.2$ の比率で分割し、 β を $\{10^{-k} \mid k = 0, 1, 2, 3, 4\}$ から探索した。LLM は gpt-3.5-turbo-0125 を使用した。LLM を用いたデータ拡張では、few-shot 学習の品質を保つため、購買ログが 3 個以上あるユーザのみをウォームスタートデータとした。

4.2 性能評価

4.2.1 明示的フィードバック

各手法の性能を MAE と Baseline (平均評価) に対する改善率で評価した。表 2 に示すように、RevAug は Music を除くすべてのデータセットで ChatGPT, Heater, Ablation を上回った。Music に注目すると、

表 2 MAE (左) と Baseline と比較した改善率 (右、% 表示)。各列で最も優れた数値は太字で示す。

	Movie	Music	Book	Grocery
Baseline	0.857	0.735	0.759	0.670
ChatGPT	0.800 (6.7)	0.719 (2.2)	0.718 (5.3)	0.705 (-5.3)
Heater	0.806 (5.9)	0.711 (3.2)	0.775 (-2.2)	0.655 (2.2)
Ablation	0.820 (4.3)	0.703 (4.4)	0.723 (4.7)	0.628 (6.1)
RevAug	0.785 (8.3)	0.714 (2.9)	0.708 (6.6)	0.626 (6.5)

表 3 nDCG@10 (左) と Baseline と比較した改善率 (右、% 表示)。各列で最も優れた数値は太字で示す。

	Movie	Music	Book	Grocery
Baseline	0.067	0.067	0.059	0.058
CLCRec	0.057 (-15.3)	0.085 (26.0)	0.062 (5.4)	0.019 (-67.0)
Ablation	0.093 (39.5)	0.080 (18.8)	0.078 (33.2)	0.054 (-7.4)
RevAug	0.123 (83.8)	0.087 (29.8)	0.080 (36.0)	0.085 (46.9)

RevAug は他の手法を上回らなかったものの、その差はわずかであった。これらの結果は、逆プロンプトによるデータ生成を通じて、RevAug が明示的フィードバックのコールドスタート推薦を高精度に解けることを示している。

4.2.2 暗黙的フィードバック

各手法の性能を nDCG@10 と Baseline (ランダムソート) に対する改善率で評価した。表 3 に示すように、RevAug はすべてのデータセットで最良の結果を達成した。Movie と Grocery では、CLCRec の改善率がマイナスとなり学習に失敗した一方で、RevAug は大幅に向上させた。これは、生成データにより教師データの不足を補えることを示唆している。また、Grocery における Ablation の改善率の低下は、データ拡張の品質が重要であることを確認させる結果となった。これらの結果から、RevAug は暗黙的フィードバックにおいてもコールドスタート推薦に効果があることが示された。

4.3 コスト分析

最後に、LLM に関連する推論コストを報告する。ユーザーの特徴量 \mathbf{f}_u は訓練段階で計算されるため、RevAug の推論コストはコールドスタートアイテムの特徴 \mathbf{f}_i の計算と、行列演算による評価または好みスコアの予測のみを含む。一方、ChatGPT は学習時のコストを必要としないが、推論段階で全ユーザーとアイテムのペアを予測するにはコストが嵩む。

表 4 では、RevAug と ChatGPT で明示的フィードバックの推論段階にかかる処理時間と RevAug のデータ拡張段階の処理時間を集計した。これによ

表 4 処理時間 (秒)

		Movie	Music	Book	Grocery
RevAug	(データ拡張)	1596	413	337	111
	推論	0.302	0.077	0.071	0.041
ChatGPT		1570	366	377	191

ると、RevAug の推論時間は ChatGPT と比較して 1000 倍以上高速であった。学習段階では、RevAug は LLM をデータ拡張のみに使用するため、LLM に関連するコストは推論対象の数に関わらず一定である。それに対して、ChatGPT は推論対象が増えるとコストも増加する。特に、全ユーザーとアイテムの組を予測する必要がある暗黙的フィードバックの場合、組み合わせ的にコストが大きくなる。例えば、Movie では明示的フィードバックのために 3069 回の推論が必要だったが (表 5)、暗黙的フィードバックの場合は 3000×100 のユーザーとアイテムの全ペアに対する推論が必要となる。

Movie での明示的フィードバック予測の場合、RevAug の学習段階での gpt-3.5-turbo-0125 による API の利用料金は 0.19 USD、ChatGPT は 0.24 USD であった。暗黙的フィードバック予測の場合、ChatGPT は単一の小規模データセットに対しても 24 USD 以上と 1.5×10^6 秒 (約 2.7 時間) のコストが必要になった。コールドスタートアイテムの数が 10 倍に増えた場合、ChatGPT の推論時間とコストも 10 倍 (240 USD ユーザーと 27 時間) となる。一方、RevAug は追加の API 料金が 0.19 USD を超えることなく、約 600 秒で高速推論が可能である。まとめると、RevAug は低コストで高精度な推論が可能である。

5 Conclusion

本論文では、ウォームスタートの訓練データが限られている状況で、コスト効率の高いコールドスタート推薦システムを強化するための新しい手法である RevAug を提案した。RevAug は LLM を用いて擬似サンプルを生成し、それらを用いて推薦システムを訓練する。数値実験により、RevAug が少ないウォームスタートデータで高精度にコールドスタート問題を解けることが検証された。また、LLM ベースの予測と比較して大幅なコスト削減が可能であった。

参考文献

- [1] Yehuda Koren, Robert M. Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. **Computer**, Vol. 42, No. 8, pp. 30–37, 2009.
- [2] Steffen Rendle, Walid Krichene, Li Zhang, and John R. Anderson. Neural collaborative filtering vs. matrix factorization revisited. In **RecSys**, pp. 240–248. ACM, 2020.
- [3] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In **RecSys**, pp. 101–109. ACM, 2019.
- [4] Maksims Volkovs, Guang Wei Yu, and Tomi Poutanen. Dropoutnet: Addressing cold start in recommender systems. In **NeurIPS**, pp. 4957–4966, 2017.
- [5] Ziwei Zhu, Shahin Sefati, Parsa Saadatpanah, and James Caverlee. Recommendation for new users and new items via randomized training and mixture-of-experts transformation. In **SIGIR**, pp. 1121–1130. ACM, 2020.
- [6] Yinwei Wei, Xiang Wang, Qi Li, Liqiang Nie, Yan Li, Xuanping Li, and Tat-Seng Chua. Contrastive learning for cold-start recommendation. In **ACM Multimedia**, pp. 5382–5390. ACM, 2021.
- [7] Zhihui Zhou, Lilin Zhang, and Ning Yang. Contrastive collaborative filtering for cold-start item recommendation. In **WWW**, pp. 928–937. ACM, 2023.
- [8] Haoyue Bai, Min Hou, Le Wu, Yonghui Yang, Kun Zhang, Richang Hong, and Meng Wang. Gorec: A generative cold-start recommendation framework. In **Multimedia**, pp. 1004–1012. ACM, 2023.
- [9] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, Hui Xiong, and Enhong Chen. A survey on large language models for recommendation. **World Wide Web (WWW)**, Vol. 27, No. 5, p. 60, 2024.
- [10] Yuhui Zhang, Hao Ding, Zeren Shui, Yifei Ma, James Zou, Anoop Deoras, and Hao Wang. Language models as recommender systems: Evaluations and limitations. In **I (Still) Can't Believe It's Not Better! NeurIPS 2021 Workshop**. OpenReview.net, 2021.
- [11] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. Chat-rec: Towards interactive and explainable llms-augmented recommender system. **CoRR**, Vol. abs/2303.14524, , 2023.
- [12] Xuansheng Wu, Huachi Zhou, Wenlin Yao, Xiao Huang, and Ninghao Liu. Towards personalized cold-start recommendation with prompts. **CoRR**, Vol. abs/2306.17256, , 2023.
- [13] Yaochen Zhu, Liang Wu, Qi Guo, Liangjie Hong, and Jundong Li. Collaborative large language model for recommender systems. In **WWW**, pp. 3162–3172. ACM, 2024.
- [14] Dairui Liu, Boming Yang, Honghui Du, Derek Greene, Neil Hurley, Aonghus Lawlor, Ruihai Dong, and Irene Li. Recprompt: A self-tuning prompting framework for news recommendation using large language models. In **CIKM**, pp. 3902–3906. ACM, 2024.
- [15] Junling Liu, Chao Liu, Renjie Lv, Kang Zhou, and Yan Zhang. Is chatgpt a good recommender? A preliminary study. **CIKM 2023 GenRec Workshop**, Vol. abs/2304.10149, , 2023.
- [16] Scott Sanner, Krisztian Balog, Filip Radlinski, Ben Wedin, and Lucas Dixon. Large language models are competitive near cold-start recommenders for language- and item-based preferences. In **RecSys**, pp. 890–896. ACM, 2023.
- [17] Jianling Wang, Haokai Lu, James Caverlee, Ed H. Chi, and Minmin Chen. Large language models as data augmenters for cold-start item recommendation. In **WWW**, pp. 726–729. ACM, 2024.
- [18] Genki Kusano. Data augmentation using reverse prompt for cost-efficient cold-start recommendation. In **RecSys**, pp. 861–865. ACM, 2024.
- [19] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In **EMNLP (1)**, pp. 6894–6910. ACM, 2021.
- [20] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In **ICML**, Vol. 119 of **Proceedings of Machine Learning Research**, pp. 1597–1607. PMLR, 2020.
- [21] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In **NeurIPS**, 2020.
- [22] Jianmo Ni, Jiacheng Li, and Julian J. McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In **EMNLP/IJCNLP (1)**, pp. 188–197. ACM, 2019.
- [23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In **NeurIPS**, pp. 8024–8035, 2019.
- [24] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In **ICLR**. OpenReview.net, 2020.

表 5 データセット統計

	Movie	Music	Book	Grocery
ユーザ	3000	573	1591	570
レコード数（訓練）	6212	1250	2829	1024
レコード数（テスト）	3096	823	841	426

図 2 では明示的フィードバック用の擬似サンプル生成に使用するプロンプト、図 3 では暗黙的フィードバック用のプロンプト、図 4 では ChatGPT を用いてユーザの評価を予測する明示的フィードバック用のプロンプトを示す。

図 2 では、調整可能な単語を黄色でハイライトしている。Movie の例として、最初の行で “Here is user rating history for {movie}” と表示しているが、他のデータセットでは “movie” を適切な用語に置き換える。また、アイテムテキストを低評価で生成する場合、最後から 2 行目の “like” を “dislike” に変更する。

図 3 では、ユーザが評価したアイテムを seen アイテムとして扱い、元の評価を 5 に変更している。ユーザが評価していないアイテムはランダムに unseen アイテムとして設定した。

Input

```
Here is user rating history for movie

## likes
- It's Complicated, Movies & TV, Movies, 3.0
- Corpse Bride, Movies & TV, Science Fiction, Animation, 5.0
- Sherlock Holmes, Movies & TV, Mystery & Suspense, 4.0

## dislikes
- Beyond a Reasonable Doubt, Movies & TV, Movies, 2.0
- Into The Woods, Movies & TV, Musicals & Performing Arts, Musicals, 2.0
- The Final Cut, Movies & TV, Science Fiction, Futuristic, 2.0

Based on above rating history, please provide around 5 characteristics of movie
this user would likely like. Format your answer in JSON as shown:
{"a movie that is xxx, xxx, xxx" : rating1, "a movie that is xxx, xxx, xxx" : rating2}.
```

Output

```
{
  "a movie that is animated, adventure, family" : 4.0,
  "a movie that is thriller, action, mystery" : 4.0,
  "a movie that is drama, romance, comedy" : 3.0,
  "a movie that is historical, war, epic" : 3.0
}
```

図 2 RevAug の明示的フィードバックに対するプロンプト

```
Here is user transaction for movie.

## seen
- It's Complicated, Movies & TV, Movies, 5.0
- Corpse Bride, Movies & TV, Science Fiction, Animation, 5.0
- Sherlock Holmes, Movies & TV, Mystery & Suspense, 5.0

## unseen
- Beyond a Reasonable Doubt, Movies & TV, Movies, 1.0
- Into The Woods, Movies & TV, Musicals & Performing Arts, Musicals, 1.0
- The Final Cut, Movies & TV, Science Fiction, Futuristic, 1.0

Based on above rating history, please provide around 5 characteristics of movie
this user would likely like. Format your answer in JSON as shown:
{"a movie that is xxx, xxx, xxx" : 5.0, "a movie that is xxx, xxx, xxx" : 5.0}.
```

図 3 RevAug の暗黙的フィードバックに対するプロンプト

```
Here is user rating history for movie

## likes
- It's Complicated, Movies & TV, Movies, 3.0
- Corpse Bride, Movies & TV, Science Fiction, Animation, 5.0
- Sherlock Holmes, Movies & TV, Mystery & Suspense, 4.0

## dislikes
- Beyond a Reasonable Doubt, Movies & TV, Movies, 2.0
- Into The Woods, Movies & TV, Musicals & Performing Arts, Musicals, 2.0
- The Final Cut, Movies & TV, Science Fiction, Futuristic, 2.0

Based on above rating history, please predict user's rating for the product:
Marvel's: The Avengers, Movies & TV, Movies

(Ratings range from 1.0 (lowest) to 5.0 (highest), the output should be like rating
value only, do not explain the reason.)
```

図 4 ChatGPT による明示的フィードバックに対するプロンプト