

콜드 스타트 아이템 추천을 위한 대규모 언어 모델 상호작용 시뮬레이터

페이란 황 제남 대학교

광저우, 중국
huangfr@jnu.edu.cn

양정항* 제남대학교 광저우

, 중국
yangzhenghang@stu2022.jnu.edu.cn

준이 장* 중국 광저

우 제남 대학교
jjy0116@stu2022.jnu.edu.cn

위안첸 베이* 중국 항저

우 저장대학교
yuanchenbei@zju.edu.cn

이지에 장 지난 대

학교 광저우, 중국
wingszhangyijie@gmail.com

하오 첸

홍콩 폴리테크닉 대학교

홍콩, 중국
sundaychenhao@gmail.com

초록

콜드 아이템 추천은 협업 필터링 모델의 오랜 과제였는데, 그 이유는 콜드 아이템의 협업 기능을 모델링하기 위한 과거 사용자 상호 작용이 부족하기 때문입니다. 콜드 아이템의 콘텐츠와 행동 패턴 사이의 격차로 인해 콜드 아이템에 대한 정확한 행동 임베딩을 생성하기가 어렵습니다. 기존의 콜드 스타트 모델은 매핑 함수를 사용하여 콜드 아이템의 콘텐츠 기능을 기반으로 가짜 행동 임베딩을 생성합니다. 그러나 이렇게 생성된 임베딩은 실제 행동 임베딩과 상당한 차이가 있어 콜드 추천 성능에 부정적인 영향을 미칩니다. 이러한 문제를 해결하기 위해 크리티코는 콘텐츠를 측면을 기반으로 사용자의 행동 패턴을 모델링하는 **LLM 인터랙션 시뮬레이터 (LLM-InS)**를 제안합니다. 이 시뮬레이터를 사용하면 추천 시스템이 각 콜드 아이템에 대한 생생한 상호작용을 시뮬레이션하고 콜드 아이템에서 웜 아이템으로 직접 변환할 수 있습니다. 특히, 사용자와 아이템의 행동 패턴을 시뮬레이션할 수 있는 맞춤형 LLM 시뮬레이터의 설계 및 훈련 프로세스에 대해 설명합니다. 또한 LLM의 시뮬레이션 성능을 최대한 활용하기 위한 효율적인 '필터링 및 정제' 접근 방식을 소개합니다. 마지막으로, 아이템의 임베딩을 업데이트하는 업데이트 방법을 제안합니다. 시뮬레이션 및 실제 상호 작용을 기반으로 추천 모델 내에서 차가운 아이템과 따뜻한 아이템에 대한 트레이닝을 통합했습니다. 실제 행동 임베딩을 사용한 광범위한 실험을 통해 우리가 제안한 모델인 LLM-InS가 콜드 스타트 아이템 추천에서 9개의 최신 콜드 스타트 방법과 3개의 LLM 모델보다 성능이 뛰어나다는 것을 입증했습니다.

키워드

한이 수수료 없이 부여됩니다. ACM이 아닌 다른 사람이 소유한 이 저작물의 구성 요소에 대한 저작권은 존중되어야 합니다. 크레딧을 첨부한 초록은 허용됩니다. 다른 방식으로 복사하거나 재게시, 서버에 게시하거나 목록에 재배포하려면 사전에 구체적인 허가 및/또는 수수료가 필요합니다. permissions@acm.org 에서 허가를 요청하세요.

컨퍼런스 약어 XX, 2018년 6월 03~5일, 뉴욕 우드스탁

© 2018 미국 컴퓨터 기계학회. ACM ISBN 978-1-4503-XXXX-X/18/06... \$15.00
<https://doi.org/XXXXXXX.XXXXXXX>

CCS 개념

• 정보 시스템 → 협업 필터링; 추천 시스템; 검색 모델 및 순위.

*이 모든 학생 저자들은 이 연구에 동등하게 기여했습니다.

영리 또는 상업적 이익을 위해 사본을 만들거나 배포하지 않고 사본에 본 고지 및 첫 페이지에 전체 인용문을 표시하는 경우 개인 또는 수업용으로 이 저작물의 전체 또는 일부를 디지털 또는 하드 카피할 수 있는 권

추천 시스템, 콜드 스타트 아이템 추천, 협업 필터링, 대규모 언어 모델

ACM 참조 형식:

페이란 황, 양정항, 장준이, 유안천 베이, 장이지예, 하오 첸. 2018. 콜드 스타트 아이템 추천을 위한 대규모 언어 모델 상호작용 시뮬레이터. *학술대회 논문집에서 권리 확인*에마이(학술대회 약어 'XX')에서 정확한 학술대회 제목을 입력해야 합니다. ACM, 미국 뉴욕, 뉴욕, 11페이지. <https://doi.org/XXXXXXX.XXXXXXX>

1 소개

수십억 개의 후보 중에서 사용자에게 가장 흥미로운 아이템을 필터링하기 위해서는 수십억 개 규모의 추천 시스템에서 협업 필터링(CF)이 필수적입니다. CF 모델[12, 13, 36, 40]은 과거 상호작용을 기반으로 각 사용자 및 항목에 대한 행동 임베딩을 학습시킬 수 있습니다. 그런 다음 추천 시스템은 사용자의 임베딩에 모든 항목 임베딩을 곱하여 각 사용자가 가장 원하는 항목을 효율적으로 필터링할 수 있습니다 [2, 5, 12, 15]. 그러나 신제품, 동영상[10, 17, 19], 라이브스트림[3, 23, 31] 등 새로운 아이템이 지속적으로 업로드되기 때문에 CF 모델은 상당한 어려움에 직면해 있습니다. 사용자 상호작용 이력이 있는 '따뜻한' 아이템과 달리, 이러한 '차가운' 아이템은 행동 임베딩을 학습할 수 있는 데이터가 부족합니다. 결과적으로 이러한 콜드 아이템은 사용자에게 효과적으로 추천할 수 없어 추천 시스템의 전반적인 생태계와 수익을 저해합니다. 따라서 콜드 아이템이 웬 아이템으로 전환하기 위해서는 가능한 한 빨리 사용자 인터랙션을 축적하도록 유도해야 합니다.

콜드 아이템에 대한 인터랙션을 생성하는 것은 어렵기 때문에 현재의 콜드 스타트 방식은 주로 '임베딩 시뮬레이션' 접근 방식에 의존합니다 [4, 6, 15, 38]. 여기에는 각 콜드 아이템의 콘텐츠 특징에 따라 임베딩을 생성하는 매핑 함수를 학습하는 과정이 포함됩니다. 딥뮤직[32]은 생성된 임베딩과 실제 동작 임베딩 간의 차이를 최소화함으로써 이를 달성합니다. 이 아이디어를 기반으로 GAR [6]은 생성적 적대적 메커니즘을 사용하여 생성된 임베딩이 실제 행동 임베딩과 동일한 분포를 갖도록 합니다. ALDI [15]는 실제 행동 임베딩을 교사로서 사용하여 생성된 임베딩에 지식을 증류하는 다른 접근 방식을 취합니다. 또한 일부 방법은 추천 시스템의 견고성을 강화하기 위해 드롭아웃 전략[30]을 활용합니다. 그러나 이러한 방법들은 여전히 매핑 기능에 의존하여

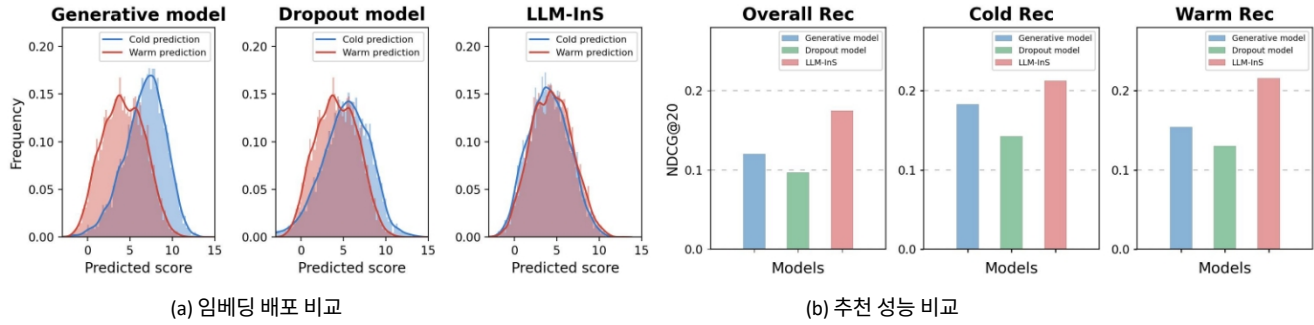


그림 1: (a) 생성/드롭아웃 임베딩 시뮬레이션 모델과 CiteULike의 LLM-InS 간의 임베딩 분포 격차를 간략하게 비교. (b) 임베딩 시뮬레이션 모델의 전체, 냉/온 추천 성능 모두에서의 시소 문제.

콘텐츠 기능을 동작 임베딩에 추가할 수 있으며, 이는 '임베딩 시뮬레이션'으로도 분류할 수 있습니다.

안타깝게도 임베딩을 시뮬레이션하는 솔루션은 실제 동작 임베딩과 생성된 임베딩 사이에 차이가 있습니다. 이는 생성된 임베딩의 분포가 실제 행동 임베딩의 분포와 크게 다르다는 것을 그림 1-(a)에서 확인할 수 있습니다. 그 결과, 그림 1-(b)에서 볼 수 있듯이 콜드 아이템의 추천은 부정적인 영향을 받게 됩니다. 이는 임베딩 시뮬레이션 모델의 한계를 잘 보여줍니다. 이러한 현상은 주로 다음과 같은 이유로 인해 발생합니다:

- 다른 임베딩 로직:** 생성된 임베딩은 콘텐츠 기능에서 생성되는 반면, 행동 임베딩은 사용자-아이템 상호 작용을 통해 학습됩니다. 이러한 근본적인 차이로 인해 임베딩 시뮬레이션 모델은 벡터 차이를 최소화할 수 있을 뿐 실제 사용자의 행동 임베딩을 사용하여 최적화되지 않습니다.
- 불충분한 콘텐츠 모델링:** 기존의 콜드 스타트 모델은 콜드 아이템의 콘텐츠 특징을 모델링하기 위해 word2vec [24] 및 MLP [28, 41]와 같은 단순한 NLP 접근 방식을 사용하는 경우가 많습니다. 그러나 이러한 단순한 NLP 접근 방식은 더 큰 매개변수를 사용하는 Berts[8, 22]와 같은 고급 모델처럼 콘텐츠의 특징을 충분히 활용할 수 없습니다. 이 두 가지 한계는 기존의 콜드 스타트 모델이 콘텐츠와 행동 사이의 간극을 효과적으로 메우는 데 방해가 됩니다.

앞서 언급한 문제를 더 잘 해결하기 위해 대규모 언어 모델(LLM)[20]을 사용하여 행동 임베딩에 대한 상호작용을 시뮬레이션하고 콘텐츠 임베딩을 더 효과적으로 모델링하는 아이디어를 고안했습니다. LLM은 자연어를 이해하고 생성하는데 상당한 발전을 이루었습니다. 그러나 콜드 스타트 추천에서의 활용은 아직 정되지 않았고 잘 정립되지 않았습니다. 콜드 스타트 항목 추천에 LLM을 적용하는 데는 몇 가지 과제가 있습니다:

- 따뜻한 아이템과 차가운 아이템을 함께 추천합니다:** 실제 추천 시스템에서는 따뜻한 아이템과 차가운 아이템을 동시에 사용자에게 추천해야 합니다. 이는 차가운 아이템도 따뜻한 아이템과 동일한 방식으로 동일한 등급

분포로 추천되어야 함을 의미합니다[6, 15].

- 행동 임베딩 생성:** 임베딩 기반 추천은 추천 시스템에서 널리 사용됩니다. 하지만 추천 시스템과 일치하는 임베딩을 생성하도록 LLM을 강제하는 것은 어려운 일입니다[1].

이 백서에서는 앞서 언급한 문제를 해결하기 위한 새로운 접근 방식을 제안합니다. 각 콜드 아이템에 대한 사용자 상호작용을 시뮬레이션하는 LLM 인터랙션 시뮬레이터(LLM-InS)를 소개합니다. 그림 1.

(a)는 콜드 아이템과 워밍 아이템의 임베딩 분포가 유사함을 보여주고, 그림 1. (b)는 콜드 아이템 추천 성능이 생성 모델이나 드롭아웃 모델과 같은 기존 콜드 스타트 모델보다 훨씬 우수하다는 것을 보여줍니다. 콘텐츠 측면에서 아이템에 대한 사용자 의도를 모델링하는 등 LLM 시뮬레이터의 설계에 대해 설명합니다. 또한, 시뮬레이터가 사용자 의도를 거칠고 세밀하게 평가할 수 있도록 LLM의 구조를 수정합니다. 또한 시뮬레이터를 효과적으로 활용하여 잠재적인 사용자 후보를 빠르게 필터링하고 구체화할 수 있는 필터링 및 구체화 접근 방식을 제시합니다. 마지막으로, 시뮬레이션과 실제 상호작용의 조합을 기반으로 항목의 임베딩을 업데이트하는 업데이트 방법을 제안하여 포괄적이고 효과적인 업데이트 프로세스를 보장합니다. 이 백서의 주요 기여는 다음과 같습니다:

- 기존의 임베딩 시뮬레이션 모델인 -10의 경우 재사용 시 성능이 크게 저하된다는 점을 강조합니다.
생성된 임베딩과 따뜻한 임베딩 사이의 눈에 띄는 차이로 인해 차가운 품목을 추천하는 데 어려움을 겪었습니다. 이러한 관찰을 바탕으로 콜드 스타트 아이템 추천을 위한 새로운 솔루션인 LLM-InS를 제안합니다.
- 대규모 언어의 구조와 교육 목표를 수정합니다.
모델을 통해 사용자 및 항목에 대한 추론 기능을 제공할 뿐만 아니라 임베딩뿐만 아니라 쌍별 상호 작용도 정확하게 시뮬레이션합니다. 콜드 아이템의 상호작용을 시뮬레이션하는 효과적인 '필터링 및 재정의' 접근 방식을 소개합니다.
- 두 가지 벤치마크 데이터 세트에 대한 광범위한 오프라인 실험을 통해 다음과 같이 입증했습니다.
LLM-InS가 9개의 최신 콜드 스타트보다 성능이 뛰어나다고 주장합니다.
그리고 전체, 차가운, 따뜻한 권장 성능의 세 가지 LLM 모델입니다.

2 관련 작업

2.1 콜드 스타트 아이템 추천

콜드 스타트 항목 추천은 새로 발생한 항목을 사용자에게 추천하는 것을 말합니다. 이러한 항목을 모델링할 행동 상호 작용이 부족하기 때문에 추천 시스템에는 장기적인 과제가 됩니다[15, 38].

현재 모델은 콜드 아이템의 콘텐츠를 임베딩에 매핑한 다음, 이를 워임 아이템에서 얻은 행동 임베딩과 정렬합니다. 정렬 방향에 따라 이러한 방법의 한 범주는 임베딩 드롭아웃 모델로, 드롭아웃 접근 방식을 통해 워임 아이템의 행동 임베딩을 콜드 아이템의 콘텐츠 생성 임베딩과 정렬하는 것을 목표로 합니다[9, 29, 33, 38, 39, 42]. 또 다른 범주는 임베딩 생성 모델로, 차가운 항목의 콘텐츠에서 매핑된 임베딩을 따뜻한 행동 임베딩의 방향으로 정렬하는 것이 목표입니다[6, 15, 25, 32]. 이는 콘텐츠 임베딩을 통해 행동 임베딩을 밀접하게 모방하려는 노력의 일환으로 수행되며, 이를 '임베딩 시뮬라토'라고 요약합니다.

2.2 LLM을 사용한 권장 사항

최근 대규모 언어 모델(LLM)은 방대 한 양의 데이터를 활용하여 정확한 문맥 인식 응답을 제공함으로써 인간과 유사한 텍스트를 이해하고 생성하는 데 탁월한 성능을 발휘하고 있습니다. 추천 시스템에는 콘텐츠에 대한 자연어 설명이 풍부하기 때문에 많은 연구가 추천 분야에서 대규모 언어 모델의 기능에 초점을 맞추기 시작했습니다[1, 18, 27, 35, 37].

특히, TALLRec[1]은 알파카 튜닝과 재튜닝이라는 두 단계의 튜닝을 포함하는 추천 작업에 LLM을 효율적으로 정렬하도록 설계된 프레임워크를 제안합니다. 이 연구는 TALLRec이 추천 기능과 강력한 교차 도메인 추천 작업을 개선한다는 것을 보여줍니다. LLMRec[37]은 세 가지 대규모 언어 모델(LLM) 기반 그래프 증강 전략을 사용하여 추천 시스템을 향상시킵니다. 이러한 전략은 사용자-아이템 상호 작용 예지 강화, 아이템 노드 속성 향상, 자연어 관점에서 사용자 노드 프로파일링 수행에 중점을 둡니다.

3 사전 예약

3.1 표기법

추천 사용자 및 항목 세트는 각각 U 와 I 로 표시됩니다. 사용자와 아이템 간의 상호작용을 나타내는 세트는 H 로 표시되며, 특히 기록 상호작용 수가 0보다 큰 아이템을 따뜻한 아이템이라고 하며 I_w 로 표시됩니다. 기록 상호작용이 0인 아이템의 경우 콜드 아이템이라고 하며 I_c 로 표시됩니다. 기록 상호작용이 있는 따뜻한 아이템의 경우 다음에서 행동 임베딩을 학습할 수 있습니다.

MF [26], LightGCN [12] 등과 같은 모델을 사용한 상호 작용을 사용합니다. 그리고

학습된 사용자 임베딩은 E_U 로 표시되며, 다음의 임베딩은

따뜻한 아이템은 E_{I_w} 으로 표시됩니다. 콜드 아이템은 상호작용 동작이 없으므로 콘텐츠 임베딩을 생성할 때는

를 사용하여 제목, 항목 설명 등의 기능에 임베딩할 수 있습니다. 이러한 콘텐츠 임베딩은 E_{I_c} 로 표시됩니다.

콜드 아이템의 핵심 문제는 이러한 아이템의 히스토리가 없다는 것입니다.

칼 상호작용. 이를 해결하기 위해 상호작용 시뮬레이터(\mathcal{I}_m)를 사용하여 시뮬레이션

을 사용하여 LLM을 학습시킵니다. 원본 LLM을 L_0 로, 훈련된 LLM을 L_t 으로 표시합니다.

3.2 문제 정의

콜드 스타트 추천. 추천 시스템의 임무는 사용자에게 항목을 추천하는 것입니다. 구체적으로, 여기에는 사용자와 항목 임베딩 간의 유사성 점수를 계산하는 작업이 포함됩니다. 그런 다음 현재 관련된 사용자와 가장 유사한 상위 항목을 추천합니다. 콜드 스타트 시나리오의 맥락에서 항목을 콜드 및 워임으로 분류할 수 있습니다. 이 시나리오에서 추천 기능을 측정하기 위해 ALDI [15]를 따라 모든 항목에 대한 추천 작업, 콜드 항목, 워임 항목의 세 가지 작업을 정의했습니다.

항목. 전체 작업 R_{all} 은 예측 점수를 기반으로 모든 항목에 대한 예측이 이루어지는 추천 작업을 말합니다.

로 표현할 수 있습니다:

$$= R_{all}P(\hat{y}_{u,i}, \forall i \in (I_w \cup I_c)). \quad (1)$$

따뜻한 추천 R_w 은 따뜻한 아이템에 대한 예측이 이루어지는 추천 작업을 말합니다.

$$w = RP(\hat{y}_{u,i}, \forall i \in I_w). \quad (2)$$

콜드 추천 R_c 은 콜드 아이템에 대한 예측이 이루어지는 추천 작업을 말합니다.

$$c = RP(\hat{y}_{u,i}, \forall i \in I_c). \quad (3)$$

여기서 P 는 사용자 및 항목에 대한 예측 점수를 나타낸 다음 예측 점수에 따라 높은 점수부터 낮은 점수까지 순위가 매겨진 상위 항목의 인덱스를 반환합니다.

상호작용 시뮬레이션. 상호작용 시뮬레이터는 주어진 사용자의 역할을 수행하여 주어진 사용자의 과거 상호작용을 수집하여 이 사용자가 다음과 같이 상호작용할 수 있는지 시뮬레이션할 수 있습니다.

콜드 아이템이 주어집니다. 구체적으로, 행동이 B 인 사용자 u 가 주어진 경우 $u =$

$\{c_{u,1}, \dots, c_{u,|B_u|}\}$, 시뮬레이터 S_{im} 은 다음과 같이 존재할 수 있습니다,

$$\hat{I}_{u,i} = S_{im}(\{c_{u,1}, \dots, c_{u,|B|}\}, c_i), \quad (4)$$

여기서 $\hat{I}_{u,i}$ 은 사용자가 다음을 수행할지에 대한 예측된 상호 작용을 나타냅니다.

u 는 아이템 i 와 상호작용합니다. 상호작용이 없는 콜드 아이템의 경우는 충분한 상호작용을 시뮬레이션해야 합니다. 따라서 각 사용자에게 대해

$u \in C_u$, 상호작용 시뮬레이터 S_{im} 은 상호작용이 발생할지 여부를 시뮬레이션해야 합니다. 모든 상호작용을 시뮬레이션한 후

호보 집합의 사용자 상호작용이 있는 것으로 시뮬레이션된 사용자를 결합하여 상호 작용 집합 I_n 을 형성합니다.

$$I = \hat{I}^n(u \in C, i \in I). \quad (5)$$

된 상호작용을 생성합니다. 각 콜드 아이템 $i \in I_c$ 에 대해 인터랙션 시뮬레이터 S_{im} 를 사용하여 해당 콜드 아이템에 대한 시뮬레이션된 인터랙션 I_n 을 생성합니다. 각 콜드 아이템의 모든 사용자에게 대한 상호작용을 계산하는 것은 비현실적입니다. 인터랙션을 빠르게 생성하려면 다음과 같이 구성된 사용자 호보 집합 $C_u, u \in U$ 를 미리 계산해야 합니다.

잠재적 상호작용 가능성이 가장 높은 사용자 비율

콜드 아이템. 상호 작용을 시뮬레이션하는 과정에서 다음이 필요합니다.

4 방법론

이 섹션에서는 LLM-InS 프레임워크를 세 부분으로 나눕니다. 먼저 자연어 이해 관점을 제공하여 상호 작용을 시뮬레이션하고 콜드 아이템에 대한 사용자 상호 작용 세트를 생성할 수 있는 LLM 시뮬레이터의 역할에 대해 간략하게 설명합니다. 그런 다음 학습된 LLM 시뮬레이터를 활용하여 각 콜드 아이템에 대한 잠재적인 사용자 인터랙션을 모방하는 계층적 인터랙션 시뮬레이터를 소개합니다. 마지막으로 시뮬레이션된 사용자 상호작용을 활용하여 콜드 아이템과 워밍 아이템을 동시에 훈련할 수 있으므로 동일한 협업 공간에서 충분한 훈련이 가능하고 콜드 임베딩과 워밍 임베딩 사이의 큰 격차를 없앨 수 있습니다. LLM-InS의 전체 아키텍처는 그림 2에 나와 있습니다.

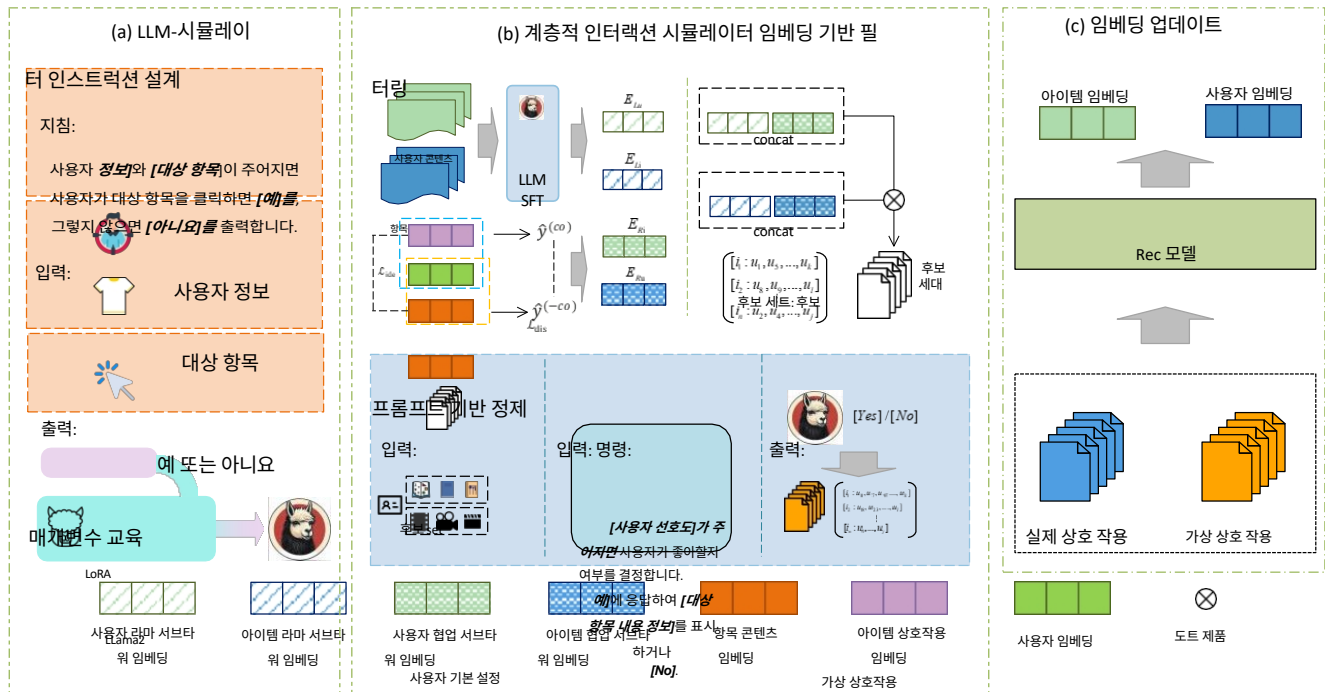


그림 2: LLM-InS의 프레임워크. 파트 (a)에서는 LLM 시뮬레이터의 전체 아키텍처를 소개합니다. 파트 (b)는 주로 계층적 인터랙션 시뮬레이터의 구조를 설명하며, 위쪽은 임베딩 기반 필터링, 나머지 부분은 프롬프트 기반 리파이닝으로 구성되어 있습니다. 파트 (c)는 임베딩 업데이트 프로세스를 간략하게 설명합니다.

4.1 LLM-시뮬레이터

기존의 콜드 스타트 방식은 콜드 아이템의 콘텐츠 정보를 모델링하기 위해 워드2벡 [24]과 같은 단순한 모델을 사용하는 경우가 많았습니다. 추천 데이터의 시맨틱 공간에만 의존하는 이러한 모델로는 충분하지 않습니다. 또한 콘텐츠 정보 모델링이 충분히 정확하고 효과적이지 못합니다. 자연어 처리 분야에서 대규모 언어 모델을 광범위하게 학습한 덕분에 프롬프트 기반 학습을 통해 항목과 사용자 간의 상호작용을 시뮬레이션하기 위해 대규모 언어 모델의 뛰어난 의미 이해도를 활용합니다.

또한, 이전 모델[21]에서는 사용자를 위한 아이템 상호작용 집합을 구성하는데 중점을 두었다면, 저희는 아이템에 대한 사용자 상호작용 집합을 구성할 것을 제안합니다. 이러한 간단한 변환을 통해 문제의 초점을 콜드 아이템에 대한 인터랙션 구성으로 전환합니다. 사용자에게 아이템을 추천하는 대신 차가운 아이템에 집중하고 모든 차가운 아이템을 최대한 포괄하는 상호작용을 구성하여 사용자가 따뜻한 아이템으로 쉽게 전환할 수 있도록 합니다. LLM을 훈련할 때는 명령어, 입력, 출력의 세 부분으로 구성하는 TALLRec[1]의 훈련 방식을 따랐습니다. 자동 회귀 손실과

파라미터 업데이트를 위해 Lora[14]를 사용하여 훈련된 LLM L_t 이 사전 정의된 출력 형식을 준수하고 시뮬레이션된 상호작용의 신뢰성을 최대화하도록 했습니다. 최적화 목표는 다음 공식으로 표현할 수 있습니다:

$$\min_{(x,y) \in Z} \sum_{t=1}^T | \Delta L_t |$$

여기서 x 는 입력, y 는 출력, Z 는 학습 집합을 나타냅니다. y_t 는 출력의 t 번째 토큰을 나타냅니다. $y_{<t}$ rep- t 번째 토큰 이전의 모든 토큰을 재전송합니다. R 은 시뮬레이터에서 생성된 상호작용의 실제 페어 비율을 나타냅니다. 자세한 내용은

LLM 시뮬레이터의 교육은 부록 A.1에 제공됩니다.

4.2 계층적 인터랙션 시뮬레이터

기존의 '임베딩 시뮬레이션' 방법과 달리, 저희는 '상호작용 시뮬레이션' 접근 방식을 제안합니다. 물론 대규모 데이터 세트에서 모든 사용자의 상호작용을 예측하는 것은 비현실적입니다. 따라서 제한된 수의 잠재적 사용자 후보를 필터링하고 선택하는 것이 보다 실용적인 접근 방식입니다. 따라서 이 문제를 해결하기 위해 임베딩 기반 필터링 시뮬레이터를 도입했습니다. 이 방법은 임베딩 수준에서 시맨틱 정보와 협업 정보를 모두 고려하기 때문에 시맨틱 공간과 협업 공간을 동시에 활용하여 후보 사용자를 선택할 수 있습니다. 이를 통해 후보 사용자 선택이 실제 시나리오와 더욱 밀접하게 일치하도록 보장합니다. 또한 LLM의 기능을 더 잘 활용하기 위해 프롬프트 기반 개선 시뮬레이터에 서명하여 디자인 프롬프트를 통해 인터랙션 시뮬레이션을 개선하고 LLM에 질문합니다. 다음은 이러한 각 측면에 대한 설명입니다.

4.2.1 임베딩 기반 필터링 시뮬레이터. 임베딩 수준에서 필터링을 더 잘 수행하기 위해 라마 서브타워를 추가로 제안합니다, 이를 사용하여 더 나은 시맨틱 공간 모델링과 Col-

$$\log((p_{LLM} + p_{Lora}) (y_t | x, y_{<t})) , \quad (6)$$

라마 서브타워. 라마 서브타워의 학습에서는 LLM-Simulator를 통해 학습된 LLM을 추천 기능이 탑재된 L_t 로 재전송합니다. 모든 데이터 세트가 에 사용자 콘텐츠 정보가 포함될 경우, 당사는 사용자가 상호작용한 일련의 아이템을 사용하여 콘텐츠 정보를 균일하게 표현합니다. 콜드 아이템과 사용자 콘텐츠 정보 모두에 대해 다음을 수행합니다.

t L에 입력하면, 다음의 모든 토큰의 임베딩을 합산하고 평균을 내서 임베딩 대표성을 계산합니다.

방정식 7에 표현된 것처럼 각 콘텐츠 정보를 입력합니다,

$$Ellm = \frac{1}{n} \sum_{i=1}^n E_{token}, \quad (7)$$

여기서 n 은 토큰 수를 나타냅니다. E_{token} 는 다음을 나타냅니다. t 에 의해 사전 훈련된 토큰을 생성합니다. 그런 다음 듀얼 타워 모델 구조를 설계합니다 BSSM[16]과 같은 구조로, 각 타워가 MLP 프로세서이며, 추천 작업에 대한 추가 사전 학습을 위해 원래 LLM 임베딩의 토큰 임베딩 $Ellm$ 을 활용합니다:

$$L_u = E_{UserTower}(Ellm), \quad (8)$$

$$L_i = E_{ItemTower}(Ellm), \quad (9)$$

여기서 E_{L_u} 와 E_{L_i} 는 각각 예상되는 사용자/아이템 임베딩입니다. tively. 프로세서 매개변수를 학습하고 최적화하기 위해 다음과 같은 작업을 별도로 수행합니다.

MLP를 통과한 후 사용자 임베딩과 아이템 임베딩 사이의 내부 곱을 계산합니다. 그런 다음 학습을 위해 BCE 손실을 사용하여 실제 클릭 상황(레이블)과 비교하여 시그모이드 함수를 통해 처리합니다. 공식적으로, 각 사용자에게

훈련 집합 T S의 항목 쌍 (u, i) , 채택된 목적 함수로 표현할 수 있습니다:

$$L_{bce} = - \frac{1}{|T|} \sum_{(u,i) \in T} y_{u,i} \log(\hat{y}_{u,i}) + (1-y_{u,i}) \log(1-\hat{y}_{u,i}), \quad (10)$$

$$\hat{y}_{u,i} = \sigma(E_{L_i} - E_{L_u}), \quad (11)$$

여기서 $\hat{y}_{u,i}$ 은 예측된 레이블이고 $y_{u,i}$ 은 실제 레이블입니다.

협업 서브타워. 라마 서브타워는 자연어 공간에서 추천 작업을 모델링하는 것을 목표로 합니다. 이를 위해 이 협업 서브타워에서 보다 합리적인 협업 임베딩을 얻는 것을 목표로 합니다. 구체적으로, 협업 정보가 있는 콜드 아이템에 대한 임베딩을 얻기 위해 워밍 아이템의 협업 정보를 활용합니다. 처음에는 기존 상호작용 정보를 기반으로 따뜻한 아이템의 임베딩을 학습합니다. 그 다음에는 손실 함수를 사용하여 임베딩 범위를 좁힙니다.

상호 작용이 없는 차가운 아이템의 임베딩과 잘 훈련된 따뜻한 아이템의 임베딩 사이의 간격은

상호작용 정보, 베이저안 개인화 순위(BPR) 손실[26]을 활용하여 협업 측면과 비협업 측면의 해당 임베딩 간의 일관성을 향상시킵니다. 따라서 해당 긍정 및 부정 표본에 대한 점수는 가능한 한 일관성이 있어야 합니다. 저희는 등급 거리 손실 L_{dis} 이를 달성하기 위해

정보), y^{co} 및 비협업 측의 내부 제품 점수(협업 정보 없음)를 나타냅니다.

점수 일관성 외에도 임베딩 거리 측면에서는 일관성 거리 손실을 통합하여 비협업 측의 양성 및 음성 샘플과 협력 측의 해당 샘플 사이의 거리를 최소화합니다. 계산 리소스를 최적화하기 위해 부정적인 항목 임베딩을 평균으로 표현하며, 그에 따른 손실은 다음과 같이 표현할 수 있습니다:

$$L_{ide} = - \sum_{i \in B} (d_i^{(co)} \ln d_i^{(co)} + (1 - d_i^{(co)}) \ln(1 - d_i^{(co)})), \quad (13)$$

$$d_i^{(co)} = \sigma(\mathbf{e}_i^\top - (\mathbf{e} - \frac{1}{|B|} \sum_{j \in B} \mathbf{e}_j)), \quad (14)$$

여기서 B 는 부정적 항목 임베딩의 배치 크기이고, \mathbf{e}_j 는 부정적 항목의 임베딩을 나타냅니다. $d_i^{(co)}$ 은 비협업 쪽에서도 동일한 작업을 수행합니다. 궁극적으로 측에서 추천 작업과 관련된 정보를 학습하기 위해 협업 정보가 없는 경우, 추천 손실로 L_{BPR} 을 사용하므로 전체 손실은 다음과 같이 표시됩니다:

$$\text{최소 } L_{\theta f} + \alpha L_{BPR} + \alpha L_{dis} + \beta L_{ide}, \quad (15)$$

여기서 α 는 추천 손실의 가중치를 나타내는 학습 가능한 파라미터이고, β 는 다른 손실의 가중치를 나타냅니다.

임베딩 병합 훈련이 끝나면 협업 서브타워 측에서 사용자와 아이템에 대한 임베딩을 얻습니다. 그런 다음 획득한 임베딩을 라마 서브타워에서 연결합니다, L_{Lu} 및 L_{Li} 으로 표시하고, 다음 공식과 같이 협업 서브타워를 기반으로 얻은 임베딩을 E_{Cu} 및 E_{Ci} 으로 표시합니다:

$$ELTi = ELi \| ECi, ELTu = ELu \| ECu. \quad (16)$$

의미 공간과 협업 공간 임베딩이 서로 간섭하지 않도록 하기 위해, 내부 곱을 수행합니다.

연산을 사용하여 임베딩을 연결할 수 있습니다. 내부 제품 연산은 같은 공간에 있는 임베딩이 곱해지도록 하고 다른 공간에 있는 임베딩은 서로 영향을 주지 않도록 합니다. 그리고

내부 제품 연산 후 얻은 임베딩은 $ELTu$ 및 $ELTi$ 으로 표시됩니다. 그 후, 각 콜드 아이템에 대해 다음 공식과 같이 내부 곱에 모든 사용자를 곱하여 상위 K 사용자 후보 C_{cold_i} 를 계산합니다:

$$I_{sim} = ELTi - E_{\mathbf{1}}^T, \quad C_{cold_i} = \text{Topk}(I_{sim}). \quad (17)$$

4.2.2 프롬프트 기반 정제 시뮬레이터. 캔-디데이트 세트 C_{cold_i} 의 사용자 세트는 프롬프트 기반 정제 시뮬레이터 중에 콜드 아이템에 대해 생성된 상호작용 세트로 활용됩니다. 행동하려면

프롬프트 기반 쿼리를 통해 추천 질문에 답하는 LLM의 기능을 개선하기 위해 프롬프트 기반 정제 시뮬레이터를 설계했습니다. 프롬프트를 구성하여 다음과 같은 상호작용을 설계합니다.

$$L = \frac{1}{n} \sum_{i=1}^n y^{(co)} - y^{(-co)} + y^{(co)} - y^{(-co)}$$

(-co)², (12) 콜 상
드 호

아 작
이 용
템 을
의
시
콘 물
텐 레
츠 이
선
정 합
보 니
와 다
사
용
자

콘
텐
츠

정
보
를

연
결
하
여

질
문
과

답
변

형
식
으
로

콜드 스타트 아이템 추천을 위한 대규모 언어 모델 상호작용 시뮬레이션

여기서 B 는 데이터 배치를 나타냅니다. u, i, j 은 각각 사용자 ID, 양수 항목 ID, 음수 항목 ID를 나타냅니다. y^{co} 는 협업 측면의 내부 제품 점수 (협업 포함)

컨퍼런스 약어 'XX', 2018년 6월 03~5일, 뉴욕 우드스탁
구체적으로는 사용자의 과거 상호작용 기록을 사용자의
콘텐츠 정보로 표시되며, 항목의 정보는 항목의 콘텐츠로 표시됩니다. 프롬프트
는 다음과 같이 공식화할 수 있습니다:

$$prompt = G_{u,i} (H_u, I | I_{tt} \in I_c), \tag{18}$$

여기서 $G_{u,i}$ 는 생성된 프롬프트 함수를 나타내고, H_u 는 사용자의 과거 상호작용 항목 콘텐츠를 다시 전송하며, l_i 는 대상 쿼리 항목을 나타냅니다. 프롬프트를 LLM에 입력한 후 출력은 다음과 같습니다.

는 이 프롬프트의 레이블로 사용되어 클릭 여부를 나타냅니다. 이 프로세스는 미세 조정 단계에서 LLM이 학습한 지식을 활용하여 임베딩 기반 필터링 시뮬레이터에서 생성된 상호 작용을 개선합니다. 사용자를 나타냅니다.

프롬프트 기반 정제 시뮬레이터 후 얻은 C_f .
프로세스는 다음 공식을 사용하여 표현할 수 있습니다:

$$C_f = L_t (\text{prompt} | A_{ns} = \text{yes}), \quad (19)$$

여기서 A_{ns} 은 LLM의 답을 나타냅니다.

4.3 임베딩 업데이트

최종 사용자 세트 C_f 를 얻은 후 모든 쿼리 아이템은 이제 다음을 얻을 수 있습니다.

사용자 setC와의 상호작용 I_{n_f} . 그들은 다음의 정보를 가지고 있습니다

LLM 시맨틱 공간과 협업 서브타워의 협업 정보로 이동합니다. 정제된 인터랙션을 과거 인터랙션 세트인 따뜻한 항목(n_w)과 병합하여 최종 인터랙션 세트 $I_{n_f} = I_{n_c} \cup I_{n_w}$. 임베딩 업데이트 중 단계에서 최종 상호작용을 균일하게 훈련했습니다 n_f . 이 접근 방식은 훈련

후 차갑고 따뜻한

항목에 더 가까이 다가갈 수 있습니다. 동시에, 우리의 설계에 따라 쿼리 아이템에 대해 생성된 인터랙션은 어느 정도 유용한 정보를 LLM에 도입했습니다. 이를 통해 상호 작용이 없는 쿼리 아이템에 대한 상호작용을 생성할 뿐만 아니라 전체 임베딩 분포를 보다 정확하게 개선할 수 있습니다. 그런 다음 MF [26], LightGCN [12], NGCF [36] 모델을 통해 상호작용 후 데이터를 처리하고, 리콜 및 NDCG 지표를 사용하여 쿼리, 쿼리 및 모든 항목에 대한 성능을 측정합니다. 조기 중지 기준은 모든 항목에서 최적의 성능을 기준으로 합니다. 이 프로세스를 통해 추천에 사용되는 최종 사용자 및 항목 임베딩이 결정됩니다.

5 실험

이 섹션에서는 다음과 같은 연구 질문에 답하기 위해 벤치마크 쿼리 스타트 추천 데이터 세트에 대한 포괄적인 실험을 수행합니다.

- **RQ1:** LLM-InS는 전체, 따뜻함, 추위에서 최신 쿼리 스타트 추천 모델보다 성능이 우수합니까?
권장 사항?
- **RQ2:** LLM-InS의 다양한 구성 요소의 효과는 무엇인가요?
- **RQ3:** LLM-InS가 현재 대표적인 추천용 LLM보다 우수한 성능을 달성할 수 있나요?

또한 부록의 내용에서 파라미터 실험과 콘텐츠 추천 및 채택 실험을 보완했습니다.

5.1 실험 설정

5.1.1 데이터 세트. 널리 사용되는 두 가지 벤치마크 데이터 세트에 대해

상호 작용. 이 연구에서는 항목의 콘텐츠 특징도 300차원 벡터를 사용하여 표현합니다. 각 데이터 세트에 대해 20%의 항목은 쿼리 시작 항목으로 지정하고, 상호작용을 쿼리 검증 세트와 테스트 세트로 나눕니다(1:1 비율). 나머지 80%의 항목에 대한 기록은 8:1:1의 비율로 훈련, 검증, 테스트 세트로 나눕니다.

5.1.2 기준선 비교. LLM-InS의 효과와 범용성을 평가하기 위해 쿼리 스타트 권장 사항 영역에서 9개의 최첨단 모델과 비교 분석을 실시했습니다. 이 비교는 두 개의 서로 다른 데이터 세트에 걸쳐 수행되었습니다. 벤치마킹한 모델에는 크게 두 가지 그룹이 있습니다: (i) 드롭아웃 기반 임베딩 시뮬레이터: **DropoutNet** [33], **MTPR** [9], **CLCRec** [38]. (ii) 제너레이티브 임베딩 시뮬레이터: **DeepMusic** [32], **MetaEmb** [25], **GPatch** [7], **GAR** [6], **UCC** [21], 그리고

ALDI [15]. 또한, 미세 조정된 기능을 검증하기 위해 포괄적인 실험을 수행합니다: **CiteULike**¹ [34]와 **Movie-Lens**² [11]를 통해 LLM-InS의 효과를 검증합니다. CiteULike는 5,551명의 사용자, 16,980개의 논문, 204,986개의 상호작용을 포함하고 있습니다. 아티클은 300차원 벡터로 표현되어 아이템 콘텐츠의 특징을 나타냅니다. 무비렌즈는 6,040명의 사용자, 3,883개의 아이템, 1,000,210개의 인터랙션으로 구성되어 있습니다.

¹ <https://github.com/js05212/citeulike-a>

² <https://grouplens.org/datasets/movielens/10m>

추천에 특화된 LLM으로 실험을 진행하기도 합니다: ChatGPT³, TALLRec [1], LLMRec [37].

5.1.3 *하이퍼파라미터/SeFing*. 임베딩 기반 필터링 단계에서는 학습률을 1e-5로 선택한 최적화 도구로 AdamW를 활용하고 각 학습 배치의 배치 크기를 128로 설정했습니다. 탐-k 값은 20으로 선택했습니다. **Prompt 기반 정제** 단계에서는 학습 속도를 5e-5로 조정했습니다. **임베딩 업데이트** 단계에서는 공식적으로 제공된 구현을 사용하여 기준선을 구현했습니다. 임베딩의 크기는 모든 모델에 대해 200으로 표준화했습니다. 학습률이 0.001인 아담 옵티마이저를 사용했으며, 검증 세트에서 NDCG를 모니터링하여 조기 종지를 적용했습니다. 공정한 비교를 위해 모든 기준선에 대해 동일한 옵션을 채택하고 각 논문에 설명된 설계를 준수했습니다.

5.1.4 *평가 지표* 저희의 평가는 전체 순위 평가 방식을 채택하여 전체, 따뜻한 추천, 차가운 추천 성능을 포괄합니다[12, 36]. 섹션 3에 제공된 정의에 따라, 저희는 추천 품질의 이 세 가지 측면을 평가합니다. 상위 랭크된 기사의 효과를 평가하기 위해 Recall@K와 NDCG@K를 주요 지표로 사용합니다. 관례에 따라 K를 20으로 설정하고 테스트 세트의 모든 사용자에게 대해 얻은 평균 값을 보고합니다. 극단적인 경우를 방지하기 위해 서로 다른 무작위 시드를 사용해 모든 실험을 5회 실행하고 평균 결과를 보고합니다.

5.2 주요 결과(RQ1)

벤치마크 데이터 세트에 대한 LLM-InS와 다른 기준선 간의 전체, 따뜻한, 차가운 추천의 성능 비교는 표 1에 나와 있습니다. LLM-InS의 일반화 가능성을 평가하기 위해 대표적인 추천 백본으로 콜드 스타트 실험을 수행했습니다: MF [26], NGCF [36], Light- GCN [12] 모델을 대표적 추천 모델로 삼아 콜드 스타트 실험을 진행합니다. 각 백본에 대해 가장 성능이 좋은 기준선과 LLM-InS를 비교하여 개선도를 계산합니다. 결과를 통해 다음과 같은 사실을 확인할 수 있습니다:

- LLM-InS는 전체, 콜드 및 웜 추천 성능 모두에서 모든 임베딩 시뮬레이션 기준 모델보다 뛰어납니다,
다양한 데이터 세트에서 지속적으로 우수성 입증

³ <https://chat.openai.com>

표 1: 세 가지 백본 모델(MF, NGCF, LightGCN)에 대한 전체, 콜드 및 웜 추천 성능 비교. 각 열의 최고 및 차순위 결과는 굵은 글꼴과 밑줄로 강조 표시되어 있습니다.

방법	전체 권장 사항 CiteULike				콜드 추천 CiteULike				따뜻한 추천 CiteULike				
	리콜	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG	
MF	백본	0.0776	0.0647	0.0932	<u>0.1377</u>	0.0056	0.0031	0.0251	0.0276	<u>0.2838</u>	<u>0.1933</u>	<u>0.2076</u>	<u>0.1819</u>
	드롭아웃넷	0.0794	0.0670	0.0646	0.1127	0.2268	0.1356	0.0671	0.0808	0.1343	0.0792	0.1391	0.1386
	MTPR	0.1060	0.0810	0.0739	0.1055	0.2496	0.1476	<u>0.0745</u>	<u>0.0811</u>	0.1728	0.0998	0.1628	0.1388
	CLCRec	0.1269	0.0992	0.0784	0.1358	0.2295	0.1347	0.0744	0.0726	0.1898	0.1167	0.1699	0.1696
	DeepMusic	0.0956	0.0789	<u>0.0933</u>	<u>0.1377</u>	0.2141	0.1262	0.0327	0.0391	<u>0.2838</u>	<u>0.1933</u>	<u>0.2076</u>	0.1819
	MetaEmb	0.0972	0.0804	<u>0.0933</u>	<u>0.1377</u>	0.2232	0.1306	0.0432	0.0468	<u>0.2838</u>	<u>0.1933</u>	<u>0.2076</u>	<u>0.1819</u>
	GPatch	0.1568	<u>0.1305</u>	0.0813	0.1253	0.2107	0.1193	0.0683	0.0704	<u>0.2838</u>	<u>0.1993</u>	<u>0.2076</u>	<u>0.1819</u>
	GAR	0.1440	0.1132	0.0462	0.0812	0.2453	0.1479	0.0348	0.0510	0.2272	0.1438	<u>0.1003</u>	<u>0.1003</u>
	ALDI	<u>0.1618</u>	0.1204	0.0914	0.1355	<u>0.2684</u>	<u>0.1550</u>	0.0431	0.0464	<u>0.2838</u>	<u>0.1993</u>	<u>0.2076</u>	<u>0.1819</u>
	LLM-InS0	<u>.2213</u>	<u>0.15990</u>	<u>.0976</u>	<u>0.14730</u>	<u>.3335</u>	<u>0.19440</u>	<u>.1225</u>	<u>0.11820</u>	<u>.3058</u>	<u>0.20610</u>	<u>.2129</u>	<u>0.1924</u>
개선	36.77%	22.52%	4.61%	6.98%	24.25%	25.41%	64.43%	45.75%	7.75%	3.41%	2.56%	5.78%	
NGCF	백본	0.1105	0.0951	<u>0.1393</u>	<u>0.2265</u>	0.0064	0.0029	0.0253	0.0275	0.2347	0.1485	<u>0.3076</u>	<u>0.2922</u>
	드롭아웃넷	0.0813	0.0656	0.1144	0.1935	0.2211	0.1278	0.0214	0.0223	0.1416	0.0842	0.2517	0.2457
	MTPR	0.1006	0.0769	0.1132	0.1818	0.2479	0.1391	0.0749	0.0894	0.1753	0.0977	0.2509	0.2329
	CLCRec	0.1201	0.0920	0.1270	0.2042	0.2093	0.1188	0.0694	0.0777	0.1886	0.1136	0.2807	0.2627
	DeepMusic	0.1269	0.1043	<u>0.1393</u>	<u>0.2265</u>	0.1980	0.1152	0.0409	0.0493	0.2347	0.1485	0.3076	0.2922
	MetaEmb	0.1119	0.0957	<u>0.1393</u>	<u>0.2265</u>	<u>0.2830</u>	<u>0.1664</u>	0.0247	0.0245	0.2347	0.1485	<u>0.3076</u>	<u>0.2922</u>
	GPatch	0.1449	0.1099	0.1381	0.2234	0.2318	0.1371	0.0589	0.0646	0.2347	0.1485	<u>0.3076</u>	<u>0.2922</u>
	GAR	0.1144	0.0909	0.0098	0.0174	0.2099	0.1251	0.0166	0.0178	0.1850	0.1143	<u>0.2788</u>	<u>0.2721</u>
	UCC	0.1094	0.0990	0.0981	0.1523	0.0019	0.0008	0.0063	0.0066	<u>0.2402</u>	<u>0.1582</u>	0.2191	0.1999
	ALDI	<u>0.1541</u>	<u>0.1141</u>	<u>0.1393</u>	0.2206	0.2466	0.1399	<u>0.1022</u>	<u>0.1113</u>	0.2347	0.1485	<u>0.3076</u>	<u>0.2922</u>
LLM-InS0 %Improv.	<u>.2211</u>	<u>0.16530</u>	<u>.1447</u>	<u>0.23720</u>	<u>.3454</u>	<u>0.20250</u>	<u>.1670</u>	<u>0.16620</u>	<u>.2992</u>	<u>0.19580</u>	<u>.3186</u>	<u>0.3052</u>	
	43.48%	44.87%	3.88%	4.72%	22.05%	21.69%	63.41%	49.33%	24.56%	23.76%	3.58%	4.45%	
LightGCN	백본	0.0812	0.0622	0.1418	<u>0.2330</u>	0.0041	0.0019	0.0177	0.0190	0.2528	0.1541	<u>0.3130</u>	<u>0.3008</u>
	드롭아웃넷	0.0883	0.0639	0.1165	0.1978	0.2309	0.1312	0.0340	0.0373	0.1175	0.0692	0.2560	0.2518
	MTPR	0.1001	0.0753	0.1011	<u>0.1551</u>	0.2585	0.1454	0.0779	0.0802	0.1753	0.0697	0.2247	0.2009
	CLCRec	0.1293	0.0965	0.1253	0.2037	0.2435	0.1425	0.0677	0.0816	0.2149	0.1302	0.2764	0.2612
	DeepMusic	0.0985	0.0745	0.1418	<u>0.2330</u>	0.2239	0.1259	0.0635	0.0719	0.2528	0.1541	0.3130	0.3008
	MetaEmb	0.0924	0.0714	0.1418	<u>0.2330</u>	0.2252	0.1295	0.0248	0.0244	0.2528	0.1541	<u>0.3130</u>	<u>0.3008</u>
	GPatch	0.1609	0.1197	0.1293	0.2106	0.2606	0.1532	0.0771	0.0760	0.2528	0.1541	<u>0.3130</u>	<u>0.3008</u>
	GAR	0.1357	0.1062	0.0106	0.0195	0.2539	0.1489	0.0110	0.0130	0.2339	0.1455	<u>0.2873</u>	<u>0.2794</u>
	UCC	0.1374	<u>0.1260</u>	0.1277	0.2020	0.0020	0.0011	0.0063	0.0073	<u>0.3002</u>	<u>0.2010</u>	0.2830	0.2641
	ALDI	<u>0.1626</u>	0.1201	<u>0.1428</u>	0.2316	<u>0.2692</u>	<u>0.1539</u>	<u>0.1229</u>	<u>0.1295</u>	0.2528	0.1541	<u>0.3130</u>	<u>0.3008</u>
LLM-InS 개선.	<u>0.2285</u>	<u>0.1747</u>	<u>0.1506</u>	<u>0.2468</u>	<u>0.3601</u>	<u>0.2126</u>	<u>0.1759</u>	<u>0.1762</u>	<u>0.3252</u>	<u>0.2156</u>	<u>0.3314</u>	<u>0.3186</u>	
	40.52%	38.65%	5.46%	5.92%	33.76%	38.14%	43.12%	36.06%	8.33%	7.26%	5.97%	5.92%	

그리고 백본. 이러한 놀라운 성과는 LLM-InS의 효과적인 상호작용 시뮬레이션이 차가운 아이템과 따뜻한 아이템에 대한 추천 기능을 동시에 크게 향상시킨 핵심 요소로 꼽을 수 있습니다.

- 제너레이티브 임베딩 시뮬레이션 모델은 일반적으로 다음을 수행합니다. 탈락자보다 따뜻하고 전반적인 권장 사항에서 더 좋습니다.
 - 침구 시뮬레이션 모델. 이는 따뜻한 임베딩과 차가운 임베딩 레이어가 임베딩 레이어를 통해 서로 접근하도록 강제하면 따뜻한 임베딩 권장 사항의 성능이 저하될 수 있음을 나타냅니다. 상호 작용 시뮬레이션

를 사용하면 차가운 상품과 따뜻한 상품 모두 통합 추천 내에서 적절하게 학습할 수 있습니다.

5.3 절제 연구(RQ2)

제안된 LLM-InS 접근 방식의 핵심 구성 요소를 검증하기 위해 제거 연구를 수행하고 그 결과를 표 2에 제시합니다.

5.3.1 필터링 단계의 효과

- 표 2의 데이터를 보면, 무작위 추출의 경우 선택하면, 이는 필터링을 수행하지 않았음을 의미하며

표 2: LLM-InS 변형을 사용한 절제 연구 결과.

변형		전체 Reco		감기 리콜		따뜻한 Recall	
		NDCG		NDCG	RealPair	NDCG	
필터링	무작위	0.1816	0.1449	0.2556	0.1628	0.0009	0.2889
	CS 미포함	0.2012	0.1543	0.3105	0.1825	0.0436	0.3067
	LS 제외	0.2070	0.1633	0.3175	0.1906	0.0401	0.3019
정제	정제 없음	0.2186	0.1668	0.3429	0.2020	0.0544	0.3124
	무작위 + 정제 + CS	0.1866	0.1507	0.2759	0.1574	0.0082	0.3189
		0.2170	0.1696	0.3313	0.1967	0.0523	0.3102
	+ 정제 없음						
	LS + 정제 없음	0.2147	0.1661	0.3271	0.1976	0.0515	0.316
LLM-InS(당사)		0.2285	0.1747	0.3601	0.2126	0.0615	0.3252
							0.2156

표 3: 권장 LLM 모델과 LLM-InS의 비교 결과.

보여줍니다. 궁극적으로 이러한 표현을 연결하여 보다 포괄적인 임베딩 표현을 얻을 수 있습니다.

작업	LLM	CiteULike		무비렌즈	
		리콜	NDCG	리콜	NDCG
전체	ChatGPT	0.2054	0.1641	0.1396	0.2267
	TALLRec	0.2141	0.1661	0.1428	0.2324
	LLMRec	0.1983	0.1535	0.1372	0.2257
	LLM-InS	0.2285	0.1747	0.1461	0.2368
	개선.	6.73%	5.18%	2.31%	1.90%
감기	ChatGPT	0.3477	0.2079	0.1480	0.1556
	TALLRec	0.3352	0.1990	0.1374	0.1379
	LLMRec	0.3453	0.2076	0.1425	0.1508
	LLM-InS	0.3601	0.2126	0.1563	0.1566
	개선.	3.57%	2.27%	5.61%	0.64%
따뜻한	ChatGPT	0.3001	0.1933	0.3074	0.2921
	TALLRec	0.3102	0.2037	0.3172	0.3016
	LLMRec	0.2613	0.1645	0.3022	0.2902
	LLM-InS	0.3252	0.2156	0.3217	0.3060
	개선.	4.84%	5.84%	1.42%	1.46%

개선했습니다. 특정 수의 사용자를 무작위로 선택하여 콜드 아이템과의 상호 작용을 구성하기만 하면 됩니다. 이 경우 리콜과 NDCG 지표가 모두 크게 감소합니다. 이는 필터링 및 정제 단계가 없으면 LLM-InS가 비교적 확실하고 신뢰할 수 있는 사용자-아이템 쌍 상호작용을 얻지 못한다는 것을 시사합니다. 이는 LLM-InS가 필터링 및 정제 단계에서 잠재적 지식을 획득하여 보다 합리적이고 신뢰할 수 있는 상호작용을 구성할 수 있음을 나타냅니다.

- 서브타워가 하나만 있는 CS 미포함 및 LS 미포함의 경우
필터링 단계에서 정제 작업이 수행되지 않고 모든 메트릭의 시나리오는 다양한 정도의 감소를 보여줍니다. 이는 하나의 서브 타워만 있으면 시맨틱 또는 협업 공간에서 부분적인 정보만 포착할 수 있으며 임베딩 벡터가 포괄적이지 않다는 것을 나타냅니다. 이는 필터링 단계에서 LLM-InS가 다양한 서브 타워로부터 서로 다른 임베딩 표현을 학습할 수 있다는 점을 정확하게

서로 다른 공간의 잠재적 정보 집합입니다. 콜드 측면의 실제 쌍 비율은 생성된 상호 작용 쌍과 실제 상호 작용 쌍의 비율을 나타내며, 다양한 하위 작업 필터링의 효과를 반영합니다.

5.3.2 정제 단계의 효율성

- 정제 단계 없이 필터링 단계만 있는 경우(예: **정제 없음**)와 비교했을 때 지표가 개선됩니다.
를 필터링 단계가 없는 것으로 변경합니다. 그러나 정제 단계가 있는 LLM-InS와 비교하면 지표가 감소합니다. 이는 정제 단계가 LLM-시뮬레이터를 통해 사용자 후보 집합을 효과적으로 정제하여 RealPair의 비율을 향상시킨다는 것을 시사합니다.
- 필터링의 세 가지 제거 실험과 비교하여 **무작위 + 정제, CS + 재 정제처럼 정제하지 않는 단계 정제, 그리고 LS + 정제 없이 정제했을 때** 표 2의 절제 메트릭은 정제 후 개선된 성능을 보여줍니다. 이는 프롬프트를 통한 시맨틱 공간 개선이 포함된 정제 단계가 전반적인 성능 향상에 효과적이라는 것을 나타냅니다. 또한 콜드 측의 리얼페어 비율의 증가는 리파이닝이 생성된 시뮬레이션 상호작용의 신뢰성을 향상시킬 수 있음을 나타냅니다.

5.4 대규모 언어 모델 비교(RQ3)

콜드 스타트 권장 설계를 위해 LLM이 LLM-InS에 포함되어 있기 때문에, LLM-InS와 현재의 최신 LLM 권장 모델도 비교합니다: ChatGPT, TALL-Rec [1], LLMRec [37]입니다. 비교 결과는 표 3에 나와 있습니다. 이 결과를 통해 LLM-InS가 일반적으로 전체, 콜드 및 웜 추천에서 현재의 모든 LLM 모델보다 우수한 성능을 보인다는 것을 알 수 있습니다. 이는 특화된 LLM-Simulator 훈련과 필터링 및 정제 단계를 통해 추천 작업을 LLM과 효과적으로 통합할 수 있음을 보여줍니다. 또한 콜드 스타트 아이템 추천에서 제안된 LLM-InS의 효율성을 강조합니다.

6 결론

이 백서에서는 콜드 스타트 추천 시나리오에서 콜드 아이템과 웜 아이템 사이의 거대한 임베딩 격차를 겨냥하여, 일반적인 콜드 스타트 모델의 한계인 '임베딩 시뮬레이터'를 활용하여 이를 극복하는 새로운 LLM-InS를 제안합니다.

- [17] 장청위안, 이허, 쟈 리, 지안 린, 레이 리, 리우준. 2019. SVD: 거의 중복에 가까운 비디오 검색을 위한 대규모 짧은 비디오 데이터 세트. *IEEE/CVF 컴퓨터 비전 국제 컨퍼런스 논문집*. 5281-5289.

패턴을 사용하여 콜드 아이템의 사실적인 상호 작용을 시뮬레이션합니다. LLM-InS의 혁신적인 워크플로는 대규모 언어 모델 기반의 필터링, 정제, 업데이트 단계로 구성되어 있습니다. 먼저 콘텐츠와 협업 공간 모두에서 사용자-아이템 관계를 공동 모델링하여 콜드 아이템의 잠재적 사용자 후보를 식별합니다. 그 다음에는 미세 조정된 LLM을 사용하여 가상 인터랙션을 완성하는 정제 프로세스가 이어집니다. 마지막으로 업데이트 단계에서 LLM-InS는 이러한 시뮬레이션 및 실제 상호 작용을 기반으로 통합 추천 모델 내에서 콜드 아이템과 워밍 아이템을 모두 학습시킵니다. 광범위한 실험을 통해 LLM-InS의 성능이 기존 방식보다 우수하다는 것이 입증되었습니다.

참고 자료

- [1] 케친 바오, 지즈 장, 양 장, 웬지 왕, 폴리 펑, 허 상난. 2023. 탈렉: 대규모 언어 모델을 추천에 맞게 조정하는 효과적이고 효율적인 튜닝 프레임워크. *arXiv 사전 인쇄물 arXiv:2305.00447* (2023).
- [2] 위안첸 베이, 하오 첸, 성위안 첸, 사오 황, 성 저우, 페이런 황. 2023. 클릭을 예측을 위한 비재귀적 클러스터 스케일 그래프 상호 작용 모델. *정보 및 지식 관리에 관한 제32회 ACM 국제 컨퍼런스*. 3748-3752.
- [3] 위안첸 베이, 하오 쉬, 성 저우, 후이첸 치, 멩디 장, 자오 리, 부 지아준. 2023. CPDG: 동적 그래프를 위한 대조적 사전 훈련 방법 신경망. *arXiv 프리프린트 arXiv:2307.02813* (2023).
- [4] 위안첸 베이, 성 저우, 차오위 탄, 하오 쉬, 하오 첸, 자오 리, 부 지아준. 2023. 비지도 그래프 이상 탐지를 위한 강화 이웃 선택. *2023 IEEE 국제 데이터 마이닝 컨퍼런스(ICDM)*. IEEE, 11-20.
- [5] 하오 첸, 위안첸 베이, 치지에 첸, 위에 쉬, 성 저우, 웬빙 황, 페이런 황, 센장 왕, 샤오 황입니다. 2024. 온라인 10억 규모 추천 시스템을 위한 매크로 그래프 신경망. *arXiv 프리프린트 arXiv:2401.14939* (2024).
- [6] 하오 첸, 제판 왕, 페이런 황, 샤오 황, 위에 쉬, 이시 린, 펑 허, 저우준 리. 2022. 콜드 스타트 아이템 추천을 위한 생성적 적대적 프레임워크. *제45회 정보 검색의 연구 및 개발에 관한 제45회 국제 ACM SIGIR 컨퍼런스 Proceedings*. 2565-2571.
- [7] 하오 첸, 제판 왕, 유예 쉬, 샤오 황, 페이런 황. 2022. GPatch: 콜드 스타트 추천을 위한 그래프 신경망 패치. *4차 워크샵에서 고차원 스페스 및 불균형 데이터에 대한 딥러닝 실습 및 이론과 KDD*.
- [8] 제이콥 데블린, 밍웨이 창, 켄트 리, 크리스티나 투타노바. 2018. Bert: 언어 이해를 위한 심층 양방향 트랜스포머의 사전 훈련. *arXiv 사전 인쇄본 arXiv:1810.04805* (2018).
- [9] 두샤오유, 시양 왕, 허상난, 리제차오, 탕진후이, 추야탕생. 2020. 콜드 스타트 멀티미디어 추천을 위한 항목 표현을 배우는 방법? *제28회 ACM 국제 멀티타임 컨퍼런스 논문집*. dia. 3469-3477.
- [10] 필리파 길, 마틴 알릿, 종평 리, 아니르반 마한티. 2007. 유튜브 트래픽 특성 분석: 옛지에서 바라본 유튜브. *인터넷 측정에 관한 제7회 ACM SIGCOMM 컨퍼런스 논문집*. 15-28.
- [11] 맥스웰 하퍼와 조셉 A 콘스탄. 2015. 영화 데이터 세트: 역사와 맥락. *대화형 지능형 시스템에 대한 Acm 트랜잭션 (TIIS)* 5, 4 (2015), 1-19.
- [12] 상난 허, 관 덩, 상 왕, 안 리, 장용동, 멩 왕. 2020. Lightgcn: 추천을 위한 그래프 컨볼루션 네트워크의 단순화 및 강화. *정보 검색 분야의 연구 및 개발에 관한 제43회 국제 ACM SIGIR 컨퍼런스 Proceedings*. 639-648.
- [13] 허 상난, 리지 라오, 장 한왕, 리창 니에, 샤 후, 차 타생. 2017. 신경 협력 필터링. *제26회 국제 월드와이드웹 컨퍼런스 논문집*. 173-182.
- [14] 에드워드 J 후, 예롱 쉰, 필립 윌리스, 제위안 알렌-주, 위안지 리, 쉰 왕, 루 왕, 웨이주 첸. 2021. Lora: 대규모 언어 모델의 저급 적응. *arXiv 프리프린트 arXiv:2106.09685* (2021).
- [15] 페이런 황, 제판 왕, 사오 황, 유펑 첸, 제타오 리, 하오 첸. 2023. 콜드 스타트 항목 추천을 위한 증류 정렬. *제46회 국제 ACM SIGIR 연구 및 개발 컨퍼런스 논문집 정보 검색*. 1147-1157.
- [16] 포센 황, 샤오동 허, 지안펑 가오, 리 덩, 알렉스 아세로, 래리 헵. 2013. 클릭 데이터를 이

- [18] 레이 리, 장용평, 두강 리우, 리 첸. 2023. 생성적 추천을 위한 대규모 언어 모델: 설문조사 및 비전 논의. *arXiv 사전 인쇄본* arXiv:2309.01157 (2023).
- [19] 형크 리아오, 에릭 맥더못, 앤드류 시니어. 2013. YouTube 동영상 전사를 위한 반지도 학습 데이터를 사용한 대규모 심층 신경망 음향 모델링. *2013 IEEE 자동 음성 인식 및 이해 워크숍에서*. IEEE, 368-373.
- [20] 장하오 린, 신이 다이, 시윤지아, 류웨이원, 보 첸, 리상양, 주첸수, 구오후평, 용유, 루이밍 탕 등 2023년. 리컴엔드 시스템이 대규모 언어 모델로부터 어떻게 이점을 얻을 수 있는가? 설문 조사. *arXiv 사전 인쇄본* arXiv:2306.05817 (2023).
- [21] 타이치 리우, 첸 가오, 젠위 왕, 동 리, 지안예 하오, 덩 진, 옹 리. 2023. 콜드 스타트 아이템 추천을 위한 불확실성 인식 일관성 학습. *제46회 정보 검색의 연구 및 개발에 관한 제46회 국제 ACM SIGIR 컨퍼런스 Proceedings* . 2466-2470.
- [22] Yinhan Liu, MyLe Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov. 2019. Roberta: 강력하게 최적화된 버트 사전 훈련 접근 방식. *arXiv 사전 인쇄본* arXiv:1907.11692 (2019).
- [23] 지충 루, 하이준 샤, 허성국, 다니엘 위그도르. 2018. 보고, 기부하고, 참여하다: 중국의 라이브 스트리밍 관행에 대한 연구. *컴퓨팅 시스템의 인적 요소에 관한 2018 CHI 컨퍼런스의 Pro- 자료*. 1-13.
- [24] 토마스 미콜로프, 카이 첸, 그렉 코라도, 제프리 딘. 2013. 벡터 공간에서 단어 표현의 효율적인 추정. *arXiv 사전 인쇄본* arXiv:1301.3781 (2013).
- [25] 판 페이양, 리 슈오카이, 상 아오, 탕 핑중, 칭 허. 2019. 콜드 스타트 광고 예열하기: ID 임베딩 학습을 통한 클릭률 예측 개선. *제42회 국제 ACM SIGIR 컨퍼런스 논문집 정보 검색의 연구 및 개발*. 695-704.
- [26] 스테펜 렌들, 크리스토프 프로이덴탈러, 제노 간트너, 라스 슈미트-티엠. 2009. BPR: 암묵적 피드백을 통한 페이지간 개인화 랭킹. *제25회 인공 지능의 불확실성에 관한 컨퍼런스 논문집* . 452-461.
- [27] 스콧 새너, 크리스티안 발로그, 필립 라들린스키, 벤 웨딘, 루카스 딕슨. 2023. 대규모 언어 모델은 언어 및 항목 기반 선호도에 대한 콜드 스타트 추천기에 가까운 경쟁력을 갖습니다. *추천 시스템에 관한 제17회 ACM 컨퍼런스 논문집* . 890-896.
- [28] 수바시 세다인, 아디타야 크리슈나 메논, 스콧 새너, 렉싱 시에. 2015. Autorec: 자동 인코더와 협업 필터링의 만남. *제24회 국제 월드와이드웹 컨퍼런스 논문집*. 111-112.
- [29] 시샤오원, 장민, 유신성, 장용평, 빈 하오, 류이첸, 마샤오핑. 2019. 누락된 콘텐츠 특징값이 있는 추천을 위한 적응형 특징 샘플링. *정보 및 지식 관리에 관한 제28회 ACM 국제 컨퍼런스* . 1451-1460.
- [30] 니티쉬 스리바스타바, 제프리 힌튼, 알렉스 크리제프스키, 일리아 수츠케베르, 루슬란 살라쿠티 디노프. 2014. 드롭아웃: 신경망의 과적합을 방지하는 간단한 방법 . *기계 학습 연구 저널* 15, 1 (2014), 1929-1958.
- [31] 존 탕, 지나 베놀리아, 코리 엠 잉크펜. 2016. 미어캣과 페리스코프: 나 스트리밍, 너 스트리밍, 라이브 스트리밍을 위한 앱 스트리밍. *2016 CHI 컴퓨팅 시스템의 인적 요소에 관한 컨퍼런스*. 4770-4780.
- [32] 아론 반 덴 오르트, 샌더 디엘레만, 벤자민 슈라우렌. 2013. 심층 콘텐츠 기반 음악 추천. *신경 정보 처리의 발전 시스템*, 26권.
- [33] 막심 볼코프스, 유광웨이, 토미 푸타넬. 2017. 드롭아웃넷: 추천 시스템에서 광고 드레싱 콜드 스타트. *신경 정보의 발전 처리 시스템*, 30권.
- [34] 왕 하오, 첸 빈이, 리 우준. 2013. 태그 추천을 위한 소셜 정규화를 통한 협업적 주제 회귀. *제23회 국제 공동 인공 지능 컨퍼런스()*에서.
- [35] 레이 왕과 임이펑. 2023. 대규모 사전 학습된 언어 모델을 사용한 제로 샷 다음 항목 추천. *arXiv 프리프린트* arXiv:2304.03153 (2023).
- [36] 상 왕, 상난 허, 행 왕, 폴리 펑, 타트생 추아. 2019. 신경 그래프 협업 필터링. *제42회 정보 검색의 연구 및 개발에 관한 제42회 국제 ACM SIGIR 컨퍼런스 논문집*. 165-174.
- [37] 웨이 웨이, 수빈 렌, 지아빈 탕, 친용 왕, 리신 수, 수치 청, 준평 왕, 다웨이 인, 차오 황. 2023. Llmrec: 추천을 위한 그래프 증강이 포함된 대규모 언어 모델. *arXiv 프리프린트* arXiv:2311.00423 (2023).
- [38] 인웨이 웨이, 시양 왕, 치 리, 리창 니에, 안 리, 쉬안핑 리, 타트생 추아. 2021. 콜드 스타트 추천을 위한 대조 학습. *제29회 ACM 멀티미디어 국제 컨퍼런스 논문집* . 5382-5390.
- [39] 샤오샤오 쉬, 첸 양, 첸 유, 지웨이 팡, 지아싱 왕, 차오성 팬, 양 허, 창핑 펑, 장강 린, 징핑 샤오. 2022. 변형 임베딩 학습으로 CTR 예측의 콜드 스타트 문제 완화 프레임워크. *ACM 웹 컨퍼런스 프로시딩* 2022. 27-35.
- [40] 유에 쉬, 하오 첸, 쟁더 덩, 유안첸 베이, 페이란 황. 2022. 추천을 위한 플랫폼-그래프 컨볼루션 네트워크. *제4회 KDD를 이용한 고차원 회소 및 불균형 데이터를 위한 딥러닝 실습 및 이론 워크숍에서*.

[41] 쉬 자오, 이 렌, 잉 두, 셴징 장, 니안 왕. 2022. 모델에 구애받지 않는 조건부 변형 자동 인코더를 통한 항목 콜드 스타트 추천 개선. *제45회 국제 ACM SIGIR 컨퍼런스 논문집 정보 검색의 연구 및 개발*: 2595-2600.

[42] 지웨이 주, 샤현 세파티, 파사 사닷파나, 제임스 카벌리. 2020. 무작위 훈련과 전문가 혼합 변환을 통한 신규 사용자 및 신규 항목에 대한 추천. *제43회 정보 검색의 연구 및 개발에 관한 제43회 국제 ACM SIGIR 컨퍼런스 논문집*: 1121-1130.

A 실험 세부 정보

A.1 LLM-시뮬레이터 교육 세트

이 연구는 사용자가 이전에 상호작용한 항목을 기반으로 대상 항목을 클릭할지 여부를 예측하는 행동을 LLM이 학습할 수 있도록 하는 것을 목표로 합니다. 데이터 세트를 구성할 때 다음과 같이 명령어, 입력 및 출력을 정렬합니다.

작업 정의, 사용자 정보, 대상 항목 콘텐츠 정보,

와 예상 결과물입니다.

교육 내용에 대해서는 다음과 같이 정의합니다: "주어진

사용자의 인터랙션 페이퍼 세트에서 사용자가 예 또는 아니요에 응답하여 대상 페이퍼를 좋아할지 여부를 결정합니다."

입력 콘텐츠에서는 자연어를 사용하여 사용자 콘텐츠 정보와 대상 항목의 콘텐츠 정보를 통합하여 추천 작업이 의존하는 콘텐츠를 설명합니다. 모든 데이터 세트에 사용자 콘텐츠 정보가 포함되어 있는 것은 아니므로 폭넓은 적용 가능성을 위해 사용자 상호 작용 세트를 구성하여 사용자 콘텐츠 정보를 표현합니다. 사용자 콘텐츠 정보의 표현으로 20개의 상호작용(또는 상호작용 길이에 따라 그 이하, 필요한 경우 1개)을 무작위로 샘플링합니다. 또한 사용자가 상호 작용했지만 사용자 콘텐츠 정보에 포함되지 않은 항목을 무작위로 긍정적인 대상 항목으로 선택합니다. 긍정적인 대상 항목의 경우 출력은 예로 설정됩니다. LLM이 단순히 Yes라고 대답하는 것을 방지하기 위해 사용자가 해당 항목과 상호작용하지 않은 부정적 목표 항목을 샘플링하여 구성하고 출력을 No로 설정합니다. 부정적 목표 항목과 긍정적 목표 항목의 비율은 1:1이며, 셔핑 후 훈련 집합으로 구성됩니다. 따라서 훈련 세트는 다음과 같이 표현할 수 있습니다:

인스트럭션: "사용자의 인터랙션 페이퍼 세트가 주어졌을 때 다음을 결정합니다.

사용자가 예 또는 아니요에 응답하여 대상 문서를 좋아할지 여부를 입력합니다." **입**

력: "사용자 선호도: [사용자가 상호작용한 논문/영화 모음], 사용자가 대상
을 좋아할지 여부

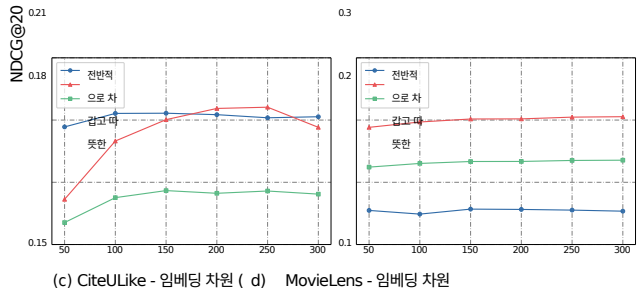
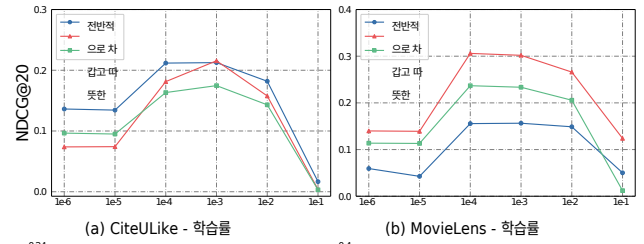
좋아/영화 [대상 좋아/영화 콘텐츠 정보?]"

출력: "예"/"아니오"

학습이 완료되고 추론이 진행되는 동안 명령어와 입력이 프롬프트에 병합되고 LLM에서 생성된 출력이 사용자-아이템 쌍의 레이블로 사용됩니다.

A.2 기준 모델에 대한 세부 정보

기준선. LLM-InS의 효과와 보편성을 평가하기 위해 두 개의 데이터 세트에서



8개의 최신 콜드 스타트 추천 모델과 비교했습니다:

- **DeepMusic**[32]은 심층 신경망을 활용하여 생성된 임베딩 간의 평균 제곱 오차 (MSE)를 최소화하고
따뜻한 임베딩.
- **MetaEmb** [25]는 메타 학습 원리를 기반으로 제너레이터를 훈련하여 빠른 수
렴을 달성합니다.
- **GAR** [6]은 따뜻한 추천 모델과 함께 생성적 적대적 프로세스에 참여하여 임
베딩을 생성합니다.

그림 3: 학습률 업데이트 및 임베딩 차원에 대한 파라미터 연구.

- **드롭아웃**[33]은 임베딩을 무작위로 폐기하여 콜드 스타트 복원력을 향상시킵니다.
 - **MTPR**[9]은 평가 탈락과 BPR 순위에 따라 사실과 다른 콜드 임베딩을 생성합니다.
 - **CLCRec** [38]은 정보 이론적 관점에서 대조 학습을 활용하여 콜드 스타트 추천에 접근합니다.
 - **GPatch** [7]는 범용 콜드 스타트 프레임워크를 도입하여 GCN 모델의 콜드 스타트 문제를 해결하고 하이브리드 및 워밍업을 촉진합니다.
 - **ALDI** [15]는 종류 학습을 사용하여 차가운 품목과 뜨거운 품목 사이의 분포 격차를 좁혀 차가운 품목에 대한 임베딩을 생성합니다.
 - **UCC** [21]는 그래프의 상관관계를 계산하여 콜드 아이템에 대한 상호 작용을 생성함으로써 콜드 스타트 문제를 해결합니다,
- 따라서 차가운 물건을 데우고 모델을 훈련시킵니다.

B 추가 실험

B.1 매개변수 연구

그림 3의 그림은 학습 속도와 임베딩 차원이라는 두 가지 하이퍼파라미터가 LightGCN 모델을 사용한 CiteULike 및 MovieLens 데이터 세트의 추천 성능에 미치는 영향을 보여줍니다. CiteULike 데이터 세트의 경우, 학습 속도를 $1e-3$ 으로 설정하고 임베딩 차원을 200으로 설정했을 때 세 가지 테스트 시나리오 모두에서 가장 좋은 결과를 얻을 수 있었습니다. 반면 MovieLens 데이터 세트의 경우, 학습 속도를 $1e-4$ 로 설정하고 임베딩 차원을 300으로 설정했을 때 세 가지 테스트 시나리오 모두에서 가장 좋은 결과를 얻었습니다. 이 결과는 학습 속도의 적당한 강도가 매우 중요하며 최적의 추천 성능을 위해 결정적인 역할을 한다는 것을 나타냅니다.

B.2 콘텐츠 추천

이 섹션에서는 콘텐츠를 대표하는 라마 서브타워를 콘텐츠 추천 모델인 딥뮤직과 ID vs Mo Rec로 교체하고 그 결과를 비교했습니다. 실험 결과는 다음과 같습니다.

표 4에 제시된 실험 결과에 따르면 당사의 접근 방식은 콜드, 웜 및 전체 시나리오에서 콘텐츠 추천 성능이 더 우수한 것으로 나타났습니다.

표 4: 콘텐츠 추천 모델과 LLM-InS 비교.

모델	전체		차갑다		따뜻함	
	리콜	NDCG	리콜	NDCG	리콜	NDCG
DeepMusic	0.2094	0.1672	0.3383	0.2073	0.2978	0.1925
IDvsMO.rec	0.1948	0.1577	0.266	0.1634	0.3207	0.2088
LLM-InS	0.2285	0.1747	0.3601	0.2126	0.3252	0.2156

B.3 추가 실험

채택률은 필터링 단계를 거친 후보 세트 중 정제 단계에서 채택된 샘플의 수를 나타냅니다. 채택률

실험 결과는 표 5에 나와 있습니다. 실험 결과 LLM-InS 프레임워크에서 채택률이 가장 높았으며, 이는 LLM-InS가 상호작용을 정확하게 시뮬레이션할 수 있음을 보여줍니다.

표 5: 채택률 비교 결과.

모델	채택률
무작위	0.0818
ALDI	0.7507
CS 미포함	0.8133
LS 제외	0.8045
LLM-InS	0.8596