

## 📌 연구 전에 세운 가정 (Assumptions)

### 1. LLM 일반화 가정

- Instruction tuning을 통해 LLM은 새로운 태스크(추천)도 빠르게 학습 가능하다.
- 즉, 언어 모델은 추천 태스크 자체를 학습한 적 없어도, 지시문을 통해 적응 가능하다고 가정.

### 2. Few-shot 학습 가능성 가정

- 추천 데이터를 충분히 모으기 어렵기 때문에, 소수의 샘플만으로도 LLM이 추천 능력을 획득할 수 있다고 가정.

### 3. 효율성 가정

- LLM은 매우 크지만, LoRA 같은 파라미터 효율적 튜닝 방식을 사용하면 GPU 자원이 제한적이어도 학습 가능하다.
- 따라서 실제 연구 환경(대학 연구실 수준)에서도 실험할 수 있다고 전제 2305.00447v3 .

### 4. 텍스트 기반 추천 가정

- 아이템에 대한 \*\*텍스트 정보(title, author, director 등)\*\*는 항상 구할 수 있다.
- 따라서 아이템 ID 기반 전통 추천 모델보다 LLM이 이 정보를 더 잘 활용할 수 있다고 가정.

## 📌 LLM을 어떻게 사용했는가

### 1. 문제 정의

- 유저의 과거 선호(좋아한 아이템, 싫어한 아이템)와 새로운 아이템의 설명을 입력.
- 출력은 "Yes/No" (사용자가 해당 아이템을 좋아할지 여부 예측) 2305.00447v3 .

### 2. 방법론 (TALLRec 프레임워크)

- **Stage 1: Alpaca Tuning**
  - Self-instruct 데이터(일반 도메인)를 활용해 LLM의 지시문 이해 및 일반화 능력 강화.
- **Stage 2: Rec-Tuning**
  - 추천 데이터를 instruction 포맷(문장형 태스크)으로 변환.
  - 예시 (논문 Table 2, p.3):
    - Input: "User's liked items: GodFather. User's disliked items: Star Wars. Target new movie: Iron Man"
    - Output: "No" 2305.00447v3 .
- **LoRA 기반 경량 튜닝 적용**
  - 모든 파라미터를 학습하지 않고 일부 저랭크 행렬만 학습 → RTX3090 24GB에서도 학습 가능 2305.00447v3 .

### 3. 실험 결과

- 소량 데이터(64샘플)에서도 기존 LLM in-context learning, 전통 추천모델보다 뛰어난 성능.
- Cross-domain (책→영화, 영화→책)에서도 높은 일반화 성능 2305.00447v3 .

## 📌 왜 LLM을 썼는가

### 1. LLM의 강점

- 방대한 사전학습으로 풍부한 지식과 언어 이해 능력 보유.
- 다양한 도메인에서 일반화 능력과 추론 능력 입증됨 2305.00447v3.
- 적절한 instruction만 주어도 새로운 태스크에 적응할 수 있음 (zero-shot, few-shot learning).

### 2. 추천 시스템에 필요한 것과 맞닿음

- 추천 시스템은 유저-아이템 상호작용이 sparse하고 cold-start가 잦음 → 일반화 능력이 필수.
- 아이템 설명(텍스트 정보)이 풍부한 경우, LLM의 언어 이해를 그대로 활용 가능.

👉 따라서 LLM을 추천 문제에 접목하면, sparse한 데이터 환경에서도 지식과 일반화 능력을 활용해 좋은 추천 성능을 낼 수 있을 것이라고 봤음.

## 3. LLM을 어떤 방법론을 사용해서 연구하였는가

- 제안 방법: TALLRec 프레임워크 (Two-stage tuning).

### 1. Alpaca Tuning

- Self-instruct 데이터(Alpaca)로 일반화 능력을 강화.

### 2. Rec-Tuning

- 추천 데이터를 instruction 포맷으로 변환해 LLM을 튜닝.
- 예:

- Input: "User liked: Godfather, disliked: Star Wars, target: Iron Man"
- Output: "No" 2305.00447v3.

- 경량 튜닝 전략 (LoRA)

- 모델 전체 파라미터를 학습하지 않고, 각 Transformer 레이어에 저랭크 행렬을 삽입하여 일부만 학습 → 원래 모델 동결, 학습량은 1/1000 수준 2305.00447v3.

- 실험:

- MovieLens (영화), BookCrossing (도서) 데이터셋.
- Few-shot (예: 64샘플)에서도 기존 LLM 기반 추천이나 전통적 추천 모델보다 성능이 높음.
- 특히 Cross-domain(책에서 학습 → 영화 추천)에서도 좋은 일반화 성능을 보임 2305.00447v3.

## 1. 어떤 문제가 발생해서 연구를 했는가

- 기존 LLM을 추천 시스템에 그대로 활용하면 성능이 낮음.
  - In-context Learning만으로는 LLM이 영화나 책 추천 시 답변을 거부하거나 항상 긍정 예측만 내놓는 문제가 발생 → 실제로는 랜덤 추측 수준( $AUC \approx 0.5$ )
- 원인:
  1. LLM이 학습된 일반 언어 처리(task)와 추천(task) 간 도메인 불일치.
  2. LLM 학습 시 추천 데이터 부족.
- 따라서 LLM이 본래 가진 지식·일반화 능력을 추천 문제에 맞게 정렬(alignment) 시켜줄 새로운 접근이 필요함

## 2. 어떤 가정을 하고 연구를 했는가

- **Instruction tuning**으로 LLM은 새로운 태스크에 빠르게 적응할 수 있다는 가정.
- 소량의 추천 데이터(few-shot)로도 LLM을 튜닝하면 기존 전통적 추천 모델보다 우수한 성능을 낼 수 있다고 가정.
- 효율성 가정: LoRA 같은 경량화 튜닝을 이용하면 대규모 GPU 없이도 LLM을 추천에 활용할 수 있음. 실제로 RTX 3090(24GB) 한 장으로 실험 가능하다고 전제