인공지능 학습 결과서

1. 개요

- **목적**
  - LLM과 CF의 서로의 단점을 상호 보완하는 시스템을 구축하고자 함
  - 대화형으로 구축해 사용자의 요구사항도 수용할 수 있음.
- **목표**
  - 이 모델을 통해 사용자가 추천을 받는 것에 대한 경험을 개선할 수 있음.

2. 데이터

**데이터 관련 항목은 다른 문서들에서 자세히 명시 현 문서에서는 간략하게 설명함

- **출처**
  - Julian McAuley LAB,University of California San Diego
  - 아마존 리뷰, 메타 데이터
- **특성**
  - json형태
  - 전체 크기 376GB → 선택한 카테고리 All Beauty 540MB
- **전처리 과정**
  - title과 description 결측치 처리
  - 리뷰수와 년도로 데이터 축소
  - 필요한 데이터 추출 및 정제 (데이터 전처리문서에서 기술)

3. 모델

- **CF : SASREC모델**

- **선택이유 :** 다른 CF모델들 과 달리 self_attention으로 문장의 연속성의 중요도 파악, 효율성 높음
- **학습과정** : 없음

- **LLM : opt- 6.7b**
  - **선택이유 :** 문장 생성형 모델 중 가벼운 축에 속하는 모델 효율성 높음, 추후에 업그레이드 가능성 있음
  - **학습과정** : 없음
- **중간 연결 모델 : A-LLMREC**
  - **선택이유 :** CF와 LLM을 이어주는 모델로 현재 있는 모델들 중 성능이 높으며 최신의 모델임.
  - **학습과정**
    - 상품을 임베딩하여 CF의 임베딩 값과 정렬해주는 학습 (stage -1)
    - stage-1에서 정렬된 임베딩을 llm의 임베딩과 정렬해주는 학습 (stage-2)

선택 이유에 대한 자료

Table 1: Overall model performance (Hit@1) over various datasets. The best performance is denoted in bold.

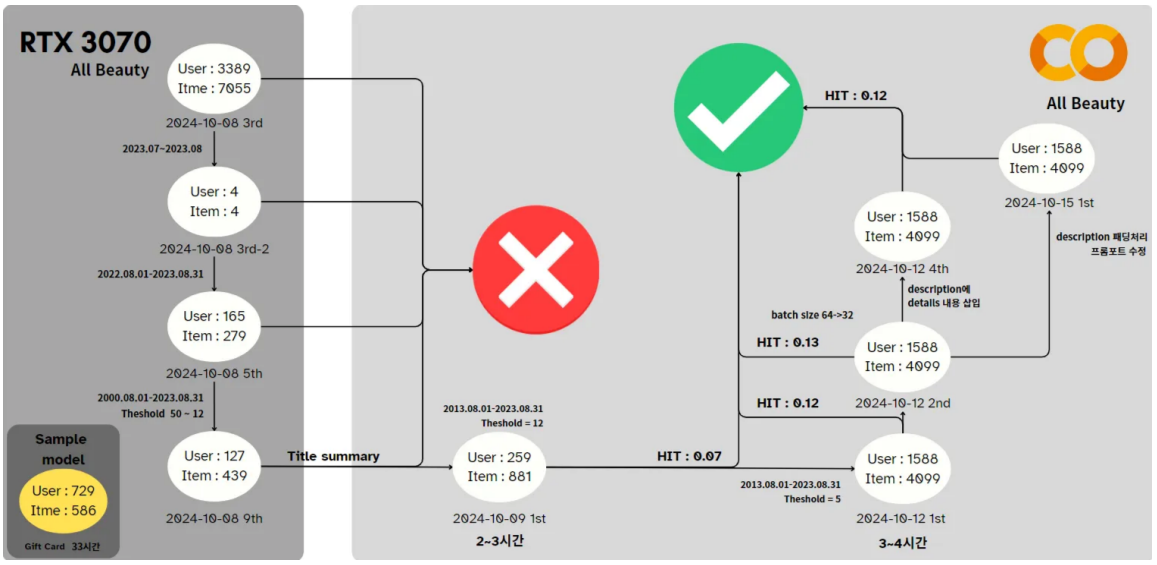
	Collaborative filtering				Modality-aware			LLM-based			
	NCF	NextItNet	GRU4Rec	SASRec	MoRec	CTRL	RECFORMER	LLM-Only	TALLRec	MLP-LLM	A-LLMRec
Movies and TV	0.4273	0.5855	0.5215	0.6154	0.4130	0.3467	0.4865	0.0121	0.2345	0.5838	**0.6237**
Video Games	0.3159	0.4305	0.4026	**0.5402**	0.4894	0.2354	0.4925	0.0168	0.4403	0.4788	0.5282
Beauty	0.2957	0.4231	0.4131	0.5298	0.4997	0.3963	0.4878	0.0120	0.5542	0.5548	**0.5809**
Toys	0.1849	0.1415	0.1673	0.2359	0.1728	0.1344	0.2871	0.0141	0.0710	0.3225	**0.3336**

- **콜드/웜 시나리오에서도 타 모델보다 성능 높음**

	Movies and TV		Video Games		Beauty	
	Cold	Warm	Cold	Warm	Cold	Warm
SASRec	0.2589	0.6787	0.1991	0.5764	0.1190	0.6312
MoRec	0.2745	0.4395	0.2318	0.4977	0.2145	0.5425
CTRL	0.1517	0.3840	0.2074	0.2513	0.1855	0.4711
RECFORMER	0.3796	0.5449	0.3039	0.5377	0.3387	0.5133
TALLRec	0.2654	0.2987	0.3950	0.4897	0.5462	0.6124
A-LLMRec	0.5714	**0.6880**	0.4263	**0.5970**	**0.5605**	**0.6414**
A-LLMRec (SBERT)	**0.5772**	0.6802	**0.4359**	0.5792	0.5591	0.6405

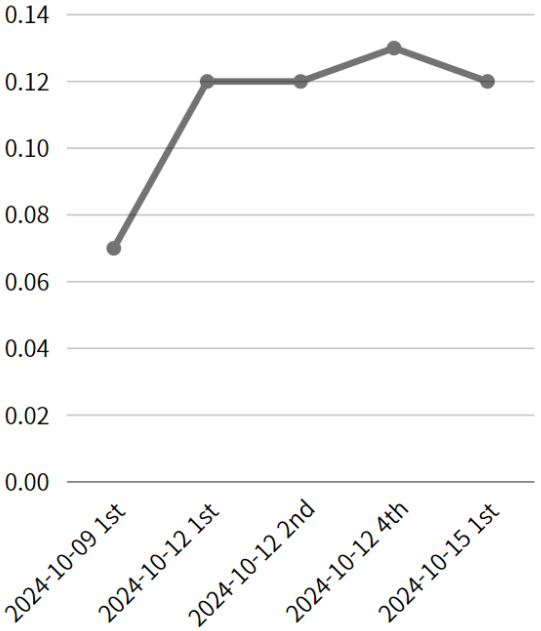
4. 결과 분석

- **모델 진행과정**



데이터 전처리를 계속 수정해가면서 학습중

- **모델 학습 결과**
  - 데이터의 범위가 늘어날 수록 성능이 좋아지는 것을 확인 함
  - 데이터의 세부적 정보를 증가시키는 방향으로 성능을 향상시킴 (description 내용 구체화)



- Hit@1 rate
  - H 모델이 사용자가 선호하는 항목을 얼마나 정확하게 1위로 추천하는지를 나타냄
  - 성공 또는 실패로 계산되며, 추천 리스트의 첫 번째 항목이 사용자가 실제로 선택한 항목과 일치하는지 여부를 측정

5. 한계점

- **모델 한계**
  - 데이터 내에서 처리하는 과정이 길어질수록(임베딩시 문장이 길면 길수록) 학습하는 과정에서 오류 발생 확률이 높아짐(메모리 부족)
    - GPT-4o-mini를 사용하여(api) title을 요약하여 학습함
  - **학습 시간 문제**
    - RTX 3070환경에서 적은 량의 데이터(50mb+2mb)도 33시간 가량 소요
    - google colab의 A100에서 500mb학습시 4시간 가량 소요
      - 여기서도 메모리 부족 문제 발생 (램 40GB)
- **개선 방향**
  - 외부 서버 활용 (더 나은 하드웨어 환경)
  - 데이터 세부사항 가공
  - LLM에서 더 나은 모델로 교체
    - 현 opt-6.7b → llama3.2 or opt-13b로 성능 개선

6. 결론 및 향후 계획

- **결론**
  - 모델은 성공적으로 학습되었고 고도화만 진행하면 될 것으로 보임
  - 추천 시스템이라는 특성상 모델의 성능이 가장 중요함
    - 콜드, 웜 시나리오를 특정하여 더 세부적인 평가를 진행해야함
- **향후 계획**
  - 모델 성능 향상을 위해 데이터 가공 및 LLM모델의 변경
  - 데이터 전처리 과정 중 리뷰 수를 변수로 두어 데이터량 증가
  - LLM을 한층 더 넣어 대화형으로 확대하여 사용자의 요구사항도 수용가능하게 개선할 예정