GPT4Rec

링크 : https://arxiv.org/abs/2304.03879

ABSTRACT

자연어 처리가 발전하면서 NLP기반의 추천시스템도 뛰어난 성능을 보임

기존 모델은 보통 아이템을 ID로만 취급하고 Discriminative modeling을 쓰기 때문에 한계가 있음

- 아이템의 컨텐츠 정보와 자연어 모델의 언어 해석 능력을 완전히 사용하지 못한다.
- 유저의 관심사를 해석해서 관심도와 다양성을 늘리지 못한다.
- 아이템이 늘어나는 것과 같은 실제 상황에 적용되지 못한다.

GPT4Rec은 검색 엔진에서 영감을 얻은 생성 프레임워크

1. 주어진 유저 히스토리의 아이템 타이틀을 활용해서 가상의 검색 쿼리를 생성
2. 가상의 쿼리를 검색해서 추천을 위한 아이템 검색

- 이 프레임워크는 유저,아이템 양 쪽의 임베딩을 언어 공간에서 학습함으로써 한계를 극복할 수 있다.
- 유저의 관심사를 잘 포착하기 위해, beam search를 사용해서 멀티 쿼리 생성을 제안한다.
    - 생성된 쿼리는 유저 관심사에 대한 해석 가능한 표현의 형태로 제공되고, cold-start 아이템을 찾게 할 수 있다.
- GPT-2 & BM25 를 사용
- beam search는 아이템 검색의 다양성과 유저의 다중 관심사 커버리지에 긍적적인 영향을 준다.

INTRODUCTION

추천 시스템은 특정 유저와 관련된 아이템에 대해 개인화된 제안을 하기 위한 정보 필터링 시스템이다.
(E-commerce, social media service 등에 적용)

자연어 처리의 발달로 많은 자연어 모델이 개인화 추천을 위해 제안됨.

최근 LLM의 성공으로 NLP 태스크나 대화에 큰 엄청낭 성능을 보였는데, 이것은 LLM 기반 추천 시스템에 박차를 가함.

특히 BERT4Rec(bi-directional self-attention 구조)는 다른 NLP 기반 모델과 시퀀셜 기반 모델들보다 뛰어난 성능을 보였다.

모델 성능이 향상됨에도 불구하고, NLP기반 모델은 전형적으로 아이템을 ID로 다루고, discriminative 모델링을 하기 때문에 아래와 같은 문제가 발생(abstract 내용)

1. 아이템의 문맥 정보나 NLP모델의 자연어 정보를 온전히 사용하지 못함
2. 아이템 목록이 바뀌거나 커지는 것에 적응하지 못한다. 이는 실제 서비스에서 중요한 문제
3. 관심사를 알아야 본질적으로 다양성이나 추천 성능을 높일 수 있을 텐데, discriminative 모델은 유저의 관심사를 설명하기 어렵다.

GPT4Rec

개인화된 추천을 제공하는 동시에 유저 관심사를 해석할 수 있음.

검색 엔진에서 영감을 얻음

1. 유저 히스토리를 프롬프트에 묶어 모델 입력으로 사용 → 가상 쿼리 생성

GPT4Rec는 강력한 생성 모델로 언어 공간에서 유저와 아이템의 임베딩 둘다를 학습 → 의미있는 정보(아이템의 타이틀과 포착된 유저의 다양한 관심사)를 활용할 수 있게 함.

유저의 다양한 관심사를 해석하고 추천 다양성을 제고하기 위해서, 쿼리 생성에 멀티 쿼리 빔 서치 기술을 사용

이 쿼리는 사람이 이해할 수 있고, 그 자체로 유저 관심사를 해석하는데 가치가 있다.

게다가 쿼리를 찾는 것은 자연스럽게 아이템 콜드 스타트와 아이템 목록 변화 이슈를 해결한다.

마지막으로 GPT4Rec은 더 진보된 LLM이나 검색엔진을 적용하는데 유연하다.

요약하자면

1. GPT4Rec(생성 프레임워크) 제안 ← 추천 태스크를 쿼리 생성 + 검색으로 다룸
2. 멀티 쿼리 빔서치를 채택하여 다양하고 해석 가능한 유저 관심사 표현을 제공

3. 두개의 데이터셋을 활용하여 실험 수행, sota보다 나은 Recall@K를 보임
4. 양적/질적 분석에서 생성 쿼리 수를 늘릴 수록, 검색 아이템의 다양성과 유저 관심사 커버리지를 높이는데 도움이 됨을 밝힘

METHODOLOGY

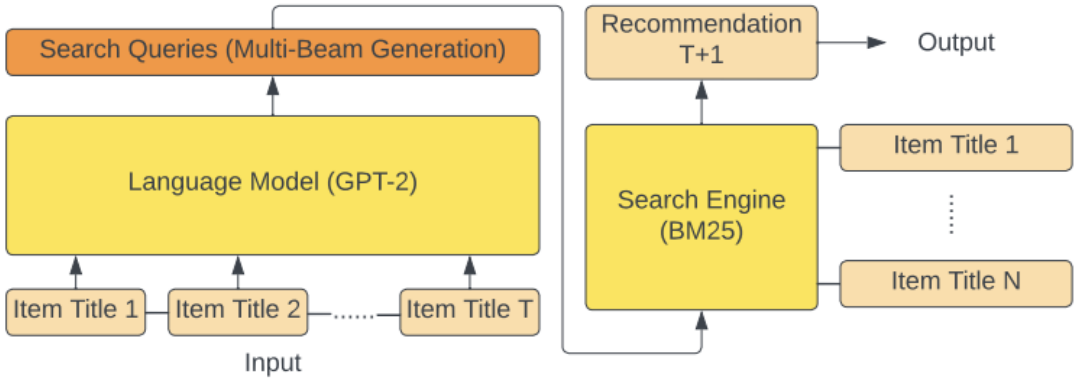


Figure 1: The architecture of the GPT4Rec framework.

- 아이템 타이틀을 프롬프트와 함께 포맷화하고, 아이템,유저 임베딩을 학습하기 위해 생성 자연어 모델을 사용
- 모델은 유저의 관심사를 표현할 수있는 여러 쿼리를 생성하고 검색 엔진에 입력해 추천을 위한 아이템 검색
- 이 논문에서는 GPT-2, BM25 사용

Query Generation with the Language Model

GPT4Rec의 첫번째 컴포넌트는 자연어 생성 모델

목표 :

- 아이템 상호작용 시퀀스로부터 자연어 공간의 유저 표현을 학습
- 유저 관심사를 표현할 수 있는 쿼리 생성

GPT-2를 파인튜닝 → 유저 관심사 및 아이템 컨텐츠 정보 포착 목적

아래와 같은 포맷의 프롬프트로 모델에 입력

Previously, the customer has bought:
<ITEM TITLE 1>. <ITEM TITLE 2>...
In the future, the customer wants to buy

이런 형식은 아이템 타이틀의 의미적 정보를 포함

유저 표현을 학습하고, 시퀀셜한 방법으로 조건부 확률에 기반한 쿼리 생성이 가능

다양한 유저의 관심사를 잘 표현하고 추천 결과의 다양성을 높이기 위해, beam search 기술을 통해 다중 쿼리 생성을 제안

beam size 를 설정하면, 빔서치 알고리즘은 generation score 함수 값이 최대가 되도록 top-m 쿼리를 업데이트 한다.

가장 중요한 것은 이러한 빔서치 전략이 다양하고, 세밀한 유저 관심사 표현을 만든다는 것이다.

Item Retrieval with the Search Engine

두번 째 컴포넌트는 discriminator로서 기능할 검색엔진

생성된 쿼리를 입력으로 하여 목록에서 매칭스코어를 기준으로 가장 관련있는 아이템을 검색한다.

매칭스코어 함수는 언어 공간에서의 유사도를 측정하고, 벡터공간에서의 내적과 같은 역할을 수행한다.

BM25 채택, 검색엔진에서 널리 사용되는 베이스라인 (k1, b 파라미터로 용어 빈도와 문서 길이를 고려함)

K : 토탈 추천 아이템 수

m : 생성 쿼리 수

각 쿼리의 검색 결과를 연결하는데에는 랭킹 기반 전략을 취한다.

- top-K/m 아이템을 가장 높은 생성 스코어를 가진 쿼리로부터의 검색 결과를 가져오고, 스코어 랭킹에 따라서 순서대로 다시 등장하지 않는 아이템을 나머지 쿼리로부터 추가한다.

이 전략은 검색된 아이템의 연관도와 다양성의 균형을 가능하게 한다.

Training Strategy

two-step 학습 절차 - 언어모델과 검색엔진을 각각 최적화, 모델 반복에 유리

언어모델 : 유저에게 1~T까지의 상호작용 시퀀스가 있다면, T-1까지를 위 프롬프트에 넣어 학습에 사용, T시점의 아이템이 fine-tune에 사용될 타이틀(가장 바람직한 결과의 가상 쿼리가 마지막 시점의 아이템 제목) 사용

BM25 : 파라미터를 그리드 서치 방법으로 생성된 쿼리가 타겟 아이템을 검색할 수 있도록 k1,b를 튜닝

EXPERIMENTS

Experiment Setup

Data

Amazon Review data[7] in categories Beauty and Electronics

아이템 타이틀에 결측치가 있거나, 길이 400을 넘는 노이즈 타이들은 제거

각 유저의 아이템 상호작용 시퀀스는 중복을 제거하고, 최대 길이를 15로 제한

전처리된 데이터의 통계 정보

Table 2: Dataset statistics.

Name	#User	#Item	#Interaction	Ave. Length	Meta Info.
Beauty	22,254	11,778	190,726	7.439	Cate.
Electronics	728,719	159,456	6,724,382	7.797	Cate., Brand

다음 아이템 추론 테스크로, 유저 시퀀스를 8:1:1로 학습,검증,테스트 셋으로 분리하고, 시퀀스의 마지막 아이템을 예측해야할 타겟으로 설정

Evaluation Metrics

Recall@K : next-item 예측의 기본 메트릭. top-K 추천에서 타겟아이템을 포함하는지를 측정, 모든 유저에 대해서 평균

Diversity@K : 아이템들의 유사하지 않음을 측정, 자카드 유사도를 사용하여 아이템의 카테고리나 브랜드 확인

$$\text{Diversity@}K = \text{Average}\left[1 - \frac{\sum_{i_1 \neq i_2} \text{Sim}(i_1, i_2)}{K(K-1)}\right],$$

Coverage@K : 추천된 아이템들이 카테고리나 브랜드 측면에서 유저 시퀀스를 얼마나 커버했는지

$$\text{Coverage@}K = \text{Average}\left[\frac{|(\bigcup_{i \in R} Cate_i) \cap (\bigcup_{i \in U} Cate_i)|}{|\bigcup_{i \in U} Cate_i|}\right],$$

Diversity는 다양한 것을 선호해서 무작위나 관련없는 것을 추천하는 것을 선호할 수 있기 때문에 coverage가 더 합리적인 지표라고 할 수 있다.

Baseline Methods

- FM-BPR
- ContentRec
- YouTubeDNN
- BERT4Rec

Inplementation Details

GPT-2

- HuggingFace,
- 117M params,
- fine-tune 20 epochs,
- Adam,
- weight decay,
- lr:0.0001,
- warm-up steps : 2000

BM25

- k in [0, 3]
- b in (0, 1)
- grid search

비교 모델

- ContentRec - using StarSpace to learn item title embeddings
- Others - using public Github repo
- embedding dimensions : 64, 128, 256
- 나머지 파라미터는 논문 참고

Quantitative Analysis

Overall Performance

Table 1: Overall performance of baseline methods and the proposed framework with different number of generated queries. The best performance is highlighted in bold font and the best baseline results is underlined.

Dataset	Recall@K	FM-BPR	ContentRec	YouTubeDNN	BERT4Rec	GPT4Rec			
						5 Queries	10 Queries	20 Queries	40 Queries
Beauty	K=5	0.0356	0.0254	<u>0.0376</u>	0.0355	**0.0653**	–	–	–
	K=10	0.0499	0.0440	<u>0.0549</u>	0.0513	0.0810	**0.1036**	–	–
	K=20	0.0716	0.0644	0.0753	<u>0.0816</u>	0.1027	0.1252	**0.1454**	–
	K=40	0.1040	0.0952	0.1066	<u>0.1161</u>	0.1297	0.1522	0.1743	**0.2040**
Electronics	K=5	0.0345	0.0241	0.0352	<u>0.0362</u>	**0.0434**	–	–	–
	K=10	0.0387	0.0307	0.0435	<u>0.0451</u>	0.0480	**0.0545**	–	–
	K=20	0.0441	0.0391	0.0539	<u>0.0573</u>	0.0524	0.0607	**0.0705**	–
	K=40	0.0505	0.0494	0.0684	<u>0.0751</u>	0.0574	0.0672	0.0788	**0.0918**

GPT4Rec와 baseline method의 Recall@K를 다양한 수의 생성쿼리로 비교한 표

gpt4rec이 가장 좋은 성능을 보임

아이템 컨텐츠 정보와 언어모델 둘다 좋은 퍼포먼스를 보이는데 중요한 재료라는 것을 보여준다.

BERT4Rec이 베이스라인 중에 최고 성능을 보였지만, 아이템을 아이디로만 사용해서 컨텐츠 정보를 온전히 사용하는데 실패했다.

반면 ContentRec은 컨텐츠 정보를 BoW 임베딩으로 사용했지만, mean풀링이 성능을 얻기에 적당하지 않았다.

Advantags of Multi-query Generation

Table1의 하삼각행렬을 보면, 행/ 열/ 대각 기준으로 증가하는 것을 볼 수 있다.

이런 추세는 멀티쿼리 빔서치 생성 전략이 추천 아이템의 연관도를 높이는데 도움이 되었다는 것을 말한다.

특히 쿼리와 검색 아이템이 1:1일때 가장 좋은 Recall@K를 확인할 수 있다.

이것은 각 쿼리가 관련있는 아이템을 찾는데 충분히 자세하다는 것을 의미한다.

Table 3: Diversity and coverage of user interests versus the number of generated queries. Highest values with regard to category/brand information are highlighted in bold font for two datasets.

Dataset		Beauty (Category)				Electronics (Category)				Electronics (Brand)			
Number of Queries		5	10	20	40	5	10	20	40	5	10	20	40
Diversity@K	K=5	**0.679**	–	–	–	**0.671**	–	–	–	**0.534**	–	–	–
	K=10	0.654	**0.716**	–	–	0.617	**0.733**	–	–	0.529	**0.643**	–	–
	K=20	0.659	0.706	**0.749**	–	0.605	0.703	**0.778**	–	0.559	0.645	**0.717**	–
	K=40	0.679	0.715	0.749	**0.783**	0.601	0.696	0.762	**0.811**	0.604	0.669	0.724	**0.767**
Coverage@K	K=5	**0.417**	–	–	–	**0.321**	–	–	–	**0.173**	–	–	–
	K=10	0.472	**0.547**	–	–	0.340	**0.425**	–	–	0.177	**0.239**	–	–
	K=20	0.535	0.602	**0.674**	–	0.364	0.447	**0.537**	–	0.184	0.245	**0.317**	–
	K=40	0.614	0.669	0.726	**0.787**	0.389	0.474	0.562	**0.653**	0.197	0.255	0.324	**0.403**

멀티 쿼리 생성 전략이 다양성과 커버리지에도 관련있는지 조사

Table 3에 보면, Diversity@K와 Coverage@K 둘다 쿼리당 한개의 아이템을 찾게 했을 때, 가장 좋은 성능을 보였다.

비슷한 증가 추세가 멀티쿼리 생성 전략이 더 이해할 수 있는 유저 관심사 표현을

반면, Diversity@K의 열을 봤을 때, 쿼리 수를 늘리지 않고 아이템 수를 늘리는 것은 다양성을 늘리는 것과는 상관 없다는 것을 보여준다.

Qualitative Analysis

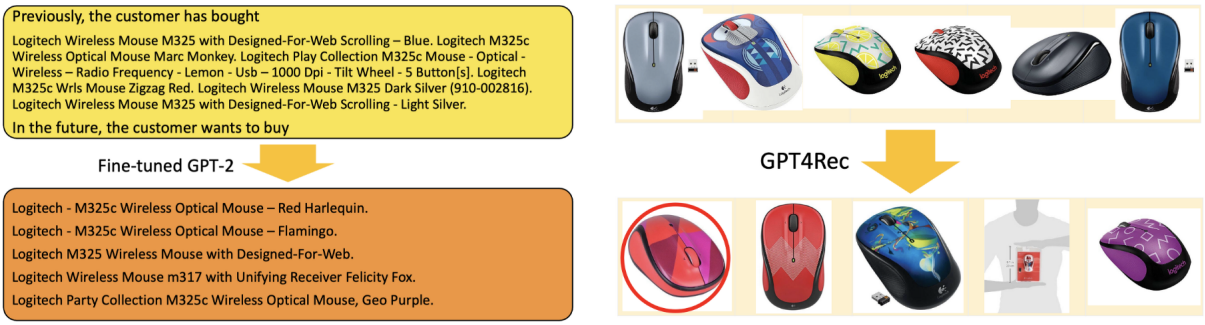
생성 쿼리의 유저 관심사 포착에 대한 효과

케이스 스터디를 통한 유저 관심사 해석의 유용성에 대해 탐색

유저가 뷰티 상품에 대해 여러 카테고리와 브랜드에서 다양한 관심사를 갖고 있을 때,

이때 GPT4Rec은 유저 히스토리 내에 없는 쿼리도 생성한다. “makeup palette”

이것은 GPT4Rec이 유저와 아이템의 상호작용과 아이템 제목의 의미 정보를 기반으로 한 아이템간의 연관성을 보착할 수 있다는 것을 의미한다.



로지텍 무선 마우스에 대한 매우 세부적인 흥미

GPT4Rec은 이 특징을 포착하기 위해 모든 생성된 쿼리에 같은 브랜드와 카테고리를 붙이고, 세부 상품은 다르게 관리했다.

위 2가지 사례를 통해 다양한 측면과 세밀한 레벨의 유저 흥미를 포착하는 GPT4Rec의 생성쿼리의 효과에 대해 설명했다.

쿼리들은 직접적으로 유저의 관심사를 해석 목적에 기여했다.

게다가 두 예시를 비교해보면 생성 쿼리의 다양성 레벨은 그들의 행동을 이해하는데 도움을 주는 유저 시퀀스의 레벨에 맞춰진다.

CONCLUSION

GPT4Rec은 개인화된 추천과 해석가능한 유저 관심사 표현을 동시에 제공한다.

GPT4Rec은 발전된 언어 모델과 아이템 컨텐츠 정보를 활용하여 높은 성능을 달성하고, 자연적으로 아이템 cold-start와 같은 문제를 해결하였다.

제안된 멀티쿼리 빔 서치 기술은 다양하고 세분화된 유저 관심사 표현을 생성하였고, 추천 결과의 관련도와 다양성을 높였다.

이 프레임워크는 더 발전된 생성 언어 모델 또는 검색 엔진, 또한 더 나은 생성이나 검색 전략으로 만들기 유연하다