BERT4Rec

BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer(2019)

BERT4Rec

개요

- 유저들에게 아이템을 추천하기 위해 유저들의 선호도를 모델링하는 것은 중요

- 기존 패러다임은 유저의 히스토리를 벡터로 인코딩, LTR 모델을 적용하여 추천
  그러나 유저의 행동시퀀스를 표현하는 것이 충분하지 않음

- 연속적인 interaction에 대해 양방향으로 context를 고려하는 방식을 고안
  -> deep bidirectional model

- 예를 들어 방에 가구를 들이기 위해 침대를 구매했는데 또 침대를 추천해주는 것은 옳지 않음

- 자연어 처리 분야에서 SoTA 모델을 기록했던 BERT모델을 이용하여
  문맥에서 단어를 예측하듯이 User에게 맞는 Item을 추천하는 방식으로 적용

BERT4Rec

기존 LTR(Left To Right) 방식의 한계

- BERT4Rec 이전에는 유저의 interaction 히스토리를 인코딩, Left To Right 방식으로 추천을 생성

- 단방향 셀프 어텐션의 한계

1. 시퀀스 내 숨겨진 영향(아이템 간 영향)을 표현하는것이 어려움

2. 엄격한 순서를 따르는 시퀀스가 아니라면 올바르지 않은 아이템 순서로 추천

따라서 User Item Interaction 데이터는 엄격한 순서가 정해져 있지 않음
단순히 어떤 유저가 어떤 아이템을 구매했는지만 알 수 있음 -> 어떤 순서로 아이템을 구매했는지 모름

유저가 구매한 아이템들끼리 어떤 영향을 끼쳤는지 표현하는 것이 제한될 뿐더러
시퀀스에 따라 올바른 추천이 생성되지 않을 가능성이 높음

BERT4Rec

BERT4Rec의 아이디어 -> Cloze task

- 전형적인 시퀀스 모델은 LTR 방식, 입력 시퀀스에서 각 위치에서 다음 아이템을 예측
  셀프 어텐션 시 target item을 참고하기 때문에 낮은 성능을 보일 수 밖에 없음
  (테스트 데이터가 훈련데이터에 새는 현상 발생)

- 이를 해결하기 위해 Cloze task 적용 -> 랜덤 마스크를 적용하여 iid를 예측
  즉, target item을 마스킹하고 information leakage를 피하고, 더 많은 훈련 샘플을 얻을 수 있음
  (데이터 증진)

- Cloze task는 꼭 최종 아이템 추천을 예측하는 것이 목적은 아님
  따라서 테스트를 할 때는 마지막에 [mask]를 추가하여 final hidden vector를 예측

BERT4Rec

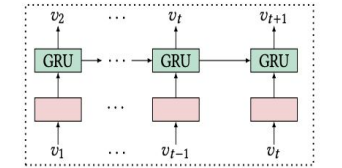
논문에서 제안하는 3가지

- 유저 행동 시퀀스에서 양방향 셀프 어텐션
- 다른 SoTA 모델과 BERT4Rec을 비교하고 양방향 셀프 어텐션의 효과를 입증
- Ablation Study

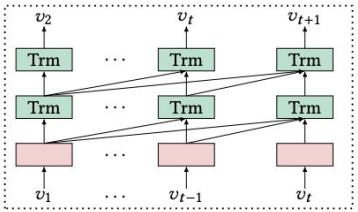
BERT4Rec

모델의 특징

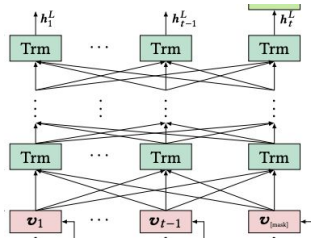
- BERT 모델에서 착안, 트랜스포머 인코더로 구성
- 이전 층으로부터 모든 포지션에 있는 표현들을 교차 연산
  cf) RNN은 이전 출력이 현재의 입력으로, SASRec은 이전 입력들과 현재 입력이 현재 출력을 생성
- CNN과 다르게 전역적인 수용영역을 얻고, RNN과 다르게 Self-Attention은 병렬화가 가능
  병렬화가 가능: 한 번에 모든 시점 입력을 연산 가능



(d) RNN based sequential recommendation methods.



(c) SASRec model architecture.



BERT4Rec

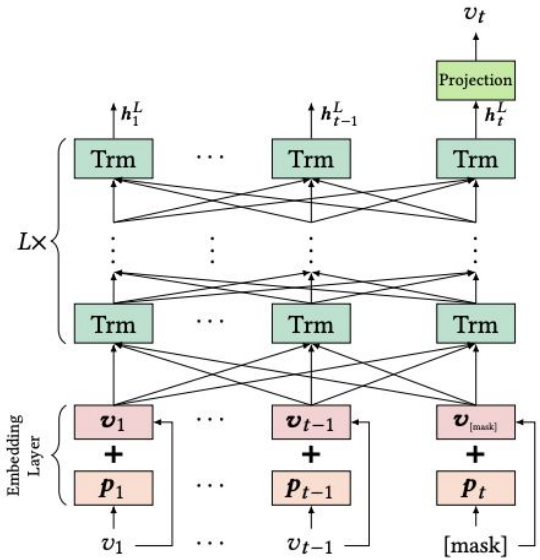
모델의 구조

- 각 아이템에 대해 아이템 임베딩과 포지션 임베딩을 더해줌
  v_t + p_t

- 이후 L개의 트랜스포머 인코더를 통과
  이 때 각 인코더는 멀티 헤드 셀프 어텐션과 PFFN을 통과

- 이 때 [mask]처리된 부분에 대해서만 아이템을 예측(양방향 어텐션
  -> Cloze task

- BERT 모델과 비교했을 때 세그먼트 임베딩이 포함이되지 않은 형타



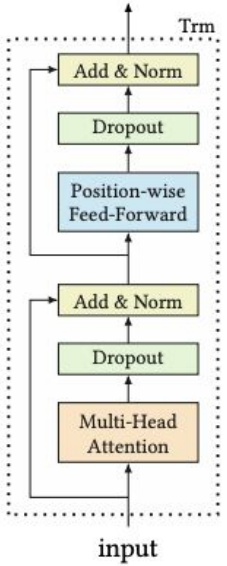
(b) BERT4Rec model architecture.

BERT4Rec

트랜스포머의 구조(보류)

- 입력 시퀀스 S에 대해 그 길이가 t

- i 번째 포지션에서 l번째 은닉층 통과, 출력 ...
- t개의



(a) Transformer Layer.

BERT4Rec

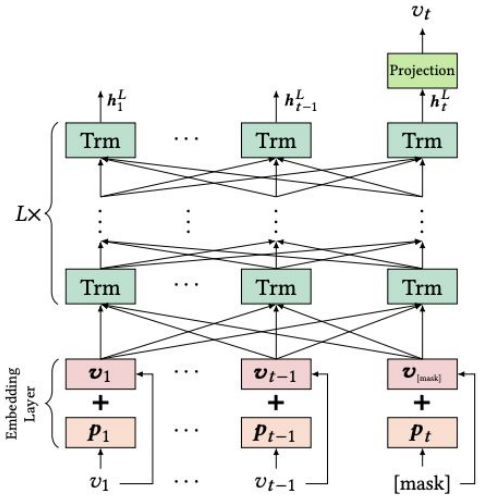
오차 함수

- [mask] 타깃에 대한 음의 로그 우도

$$\mathcal{L} = \frac{1}{|S_u^m|} \sum_{v_m \in S_u^m} -\log P(v_m = v_m^* | S'_u)$$

- -log(P(v_m = v_m*|S'_u)가 작아지는 방향으로 학습

- 즉, 유저 히스토리(S_u')이 주어졌을 때 [mask]에서 예측한 아이템(v_m)이 실제 아이템(v_m*)과 동일할 확률이 커지는 방향으로 학습



(b) BERT4Rec model architecture.

BERT4Rec

테스트

- 궁극적으로 가장 마지막 시점에서의 아이템을 예측하는 것이 목적
- 따라서 테스트를 할 때에는 유저 시퀀스에서 가장 마지막 아이템을 [mask] 토큰을 부여
- [mask]의 위치에서 추천 아이템을 생성, 추천 성능 측정

BERT4Rec

임베딩 벡터 차원의 영향

-

BERT4Rec

마스킹 비율의 영향

-

BERT4Rec

시퀀스 길이의 영향

-

BERT4Rec

Ablation Study

-

BERT4Rec

BERT4Rec의 성능

-