BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer

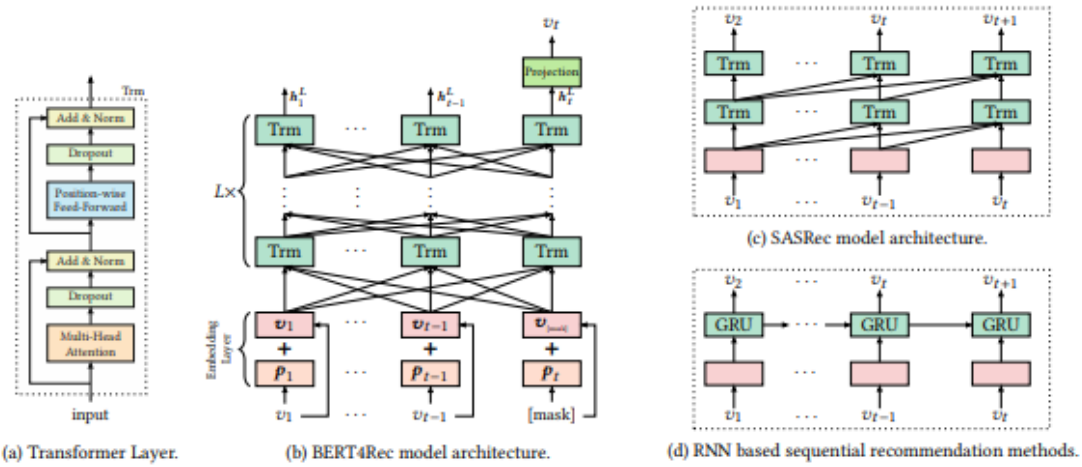
ABSTRACT

과거의 추천 시스템은 sequential neural networks을 사용해 사용자의 과거 상호작용을 왼쪽에서 오른쪽으로 단방향적으로 인코딩했다. 그러나, 시퀀셜 단방향적 모델은 과거의 정보만을 사용하기 때문에 사용자 행동 시퀀스의 숨겨진 의미를 제한하고 가끔 실용적이지 않다. 따라서, 사용자의 행동 시퀀스를 모델링하기 위해 deep bidirectional self-attention을 사용한 BERT4Rec을 제안한다. 이는 양방향이기 때문에 사용자의 과거 행동의 왼쪽과 오른쪽이 결합된다.

INTRODUCTION

전통적인 시퀀셜 추천 모델은 왼쪽에서 오른쪽으로 훈련시켜 다음 아이템을 예측하는 방식이다. 양방향 모델은 왼쪽과 오른쪽을 조인하기 때문에 정보 부족을 초래할 수 있으며 각 아이템이 간접적으로 타겟 아이템을 볼 수 있다. 이를 해결하기 위해 Cloze task를 활용한다. 랜덤하게 아이템을 마스킹하고 주위 문맥에 기반해 마스킹된 아이템의 ids를 예측한다. 게다가 Cloze objective는 더 많은 샘플을 생산해 트레이닝을 효과적으로 한다. 그러나 Cloze task는 최종 task와 일치하지 않는다. 이를 수정하기 위해 테스트를 하면서 인풋 시퀀스 마지막에 "mask" 토큰을 추가하여 예측할 아이템을 나타낸다.

BERT4REC



(a) Transformer Layer. (b) BERT4Rec model architecture. (c) SASRec model architecture. (d) RNN based sequential recommendation methods.

v_t는 사용자가 t 시점에 interaction한 아이템 v이다.

임베딩 layer의 경우 아이템의 정보와 아이템의 위치 정보를 더하고 최종 예측 목표인 Pt에 마스킹한 것을 입력으로 넣는다. 이때, 유저의 시퀀스 길이가 전체 시퀀스 길이보다 크면 잘라내고 작으면 제로 패딩한다. Transformer Layer의 경우 Multihead Attention, Point-Wise Feed Forward 사용하며 레이어 수만큼 반복 연산한다. (d)의 RNN과 같이 순차적으로 학습이 진행되는 것이 아니라 병렬적으로 학습한다. 즉, 과거 layer의 모든 정보를 반영해 학습한다. 마지막으로 Output Layer에서는 final output을 softmax해 mask 토큰의 확률값을 구한다. 그리고 mask 아이템과 실제 mask 아이템과 비교하여 negative sampling 방식으로 weight를 업데이트하며 학습을 진행한다.

EXPERIMENTS Results

Datasets	Metric	POP	BPR-MF	NCF	FPMC	GRU4Rec	GRU4Rec$^+$	Caser	SASRec	BERT4Rec	Improv.
Beauty	HR@1	0.0077	0.0415	0.0407	0.0435	0.0402	0.0551	0.0475	0.0906	**0.0953**	5.19%
	HR@5	0.0392	0.1209	0.1305	0.1387	0.1315	0.1781	0.1625	0.1934	**0.2207**	14.12%
	HR@10	0.0762	0.1992	0.2142	0.2401	0.2343	0.2654	0.2590	0.2653	**0.3025**	14.02%
	NDCG@5	0.0230	0.0814	0.0855	0.0902	0.0812	0.1172	0.1050	0.1436	**0.1599**	11.35%
	NDCG@10	0.0349	0.1064	0.1124	0.1211	0.1074	0.1453	0.1360	0.1633	**0.1862**	14.02%
	MRR	0.0437	0.1006	0.1043	0.1056	0.1023	0.1299	0.1205	0.1536	**0.1701**	10.74%
Steam	HR@1	0.0159	0.0314	0.0246	0.0358	0.0574	0.0812	0.0495	0.0885	**0.0957**	8.14%
	HR@5	0.0805	0.1177	0.1203	0.1517	0.2171	0.2391	0.1766	0.2559	**0.2710**	5.90%
	HR@10	0.1389	0.1993	0.2169	0.2551	0.3313	0.3594	0.2870	0.3783	**0.4013**	6.08%
	NDCG@5	0.0477	0.0744	0.0717	0.0945	0.1370	0.1613	0.1131	0.1727	**0.1842**	6.66%
	NDCG@10	0.0665	0.1005	0.1026	0.1283	0.1802	0.2053	0.1484	0.2147	**0.2261**	5.31%
	MRR	0.0669	0.0942	0.0932	0.1139	0.1420	0.1757	0.1305	0.1874	**0.1949**	4.00%
ML-1m	HR@1	0.0141	0.0914	0.0397	0.1386	0.1583	0.2092	0.2194	0.2351	**0.2863**	21.78%
	HR@5	0.0715	0.2866	0.1932	0.4297	0.4673	0.5103	0.5353	0.5434	**0.5876**	8.13%
	HR@10	0.1358	0.4301	0.3477	0.5946	0.6207	0.6351	0.6692	0.6629	**0.6970**	4.15%
	NDCG@5	0.0416	0.1903	0.1146	0.2885	0.3196	0.3705	0.3832	0.3980	**0.4454**	11.91%
	NDCG@10	0.0621	0.2365	0.1640	0.3439	0.3627	0.4064	0.4268	0.4368	**0.4818**	10.32%
	MRR	0.0627	0.2009	0.1358	0.2891	0.3041	0.3462	0.3648	0.3790	**0.4254**	12.24%
ML-20m	HR@1	0.0221	0.0553	0.0231	0.1079	0.1459	0.2021	0.1232	0.2544	**0.3440**	35.22%
	HR@5	0.0805	0.2128	0.1358	0.3601	0.4657	0.5118	0.3804	0.5727	**0.6323**	10.41%
	HR@10	0.1378	0.3538	0.2922	0.5201	0.5844	0.6524	0.5427	0.7136	**0.7473**	4.72%
	NDCG@5	0.0511	0.1332	0.0771	0.2239	0.3090	0.3630	0.2538	0.4208	**0.4967**	18.04%
	NDCG@10	0.0695	0.1786	0.1271	0.2895	0.3637	0.4087	0.3062	0.4665	**0.5340**	14.47%
	MRR	0.0709	0.1503	0.1072	0.2273	0.2967	0.3476	0.2529	0.4026	**0.4785**	18.85%

모든 데이터셋과 Metric에서 다른 모델보다 성능이 좋은 것을 확인할 수 있다. 또한 기존의 단방향 모델에 비해 좋은 성능을 보이는 것을 알 수 있다.