BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer

Authors: Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang

Докладчик: Андрей Семенов

#ods_recommender_systems

BERT4Rec: Sequential **recommendation** with bidirectional encoder representations from **transformer**

F Sun, J Liu, J Wu, C Pei, X Lin, W Ou… - Proceedings of the 28th …, 2019 - dl.acm.org

… Here, we introduce a new sequential **recommendation** model … from **Transformers** to a new task, sequential **Recommendation**. It is built upon the popular self-attention layer, "**Transformer** …

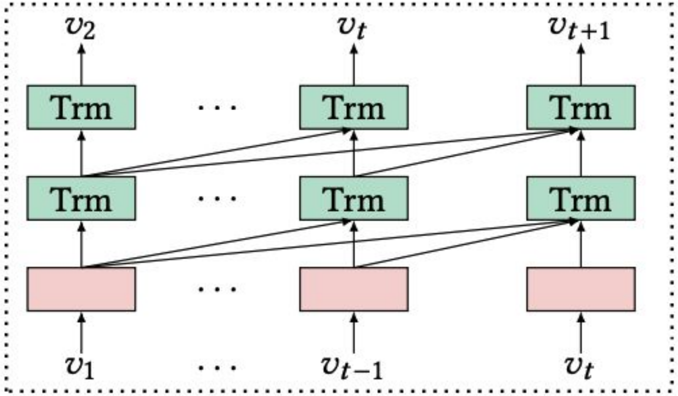
Save Cite Cited by 894 Related articles All 9 versions

INTRODUCTION

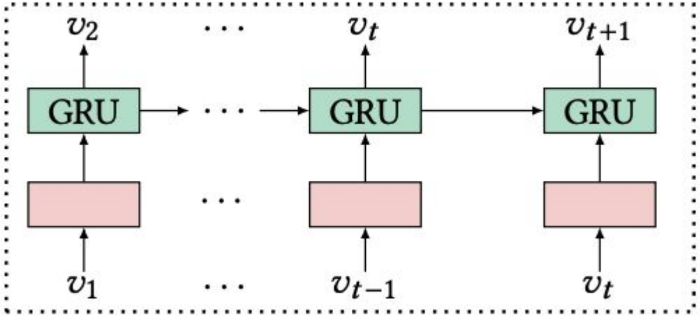
Unidirectional architectures:

a) restrict the power of hidden representation in users' behavior sequences;

b) often assume a rigidly ordered sequence which is not always practical.



(c) SASRec model architecture.

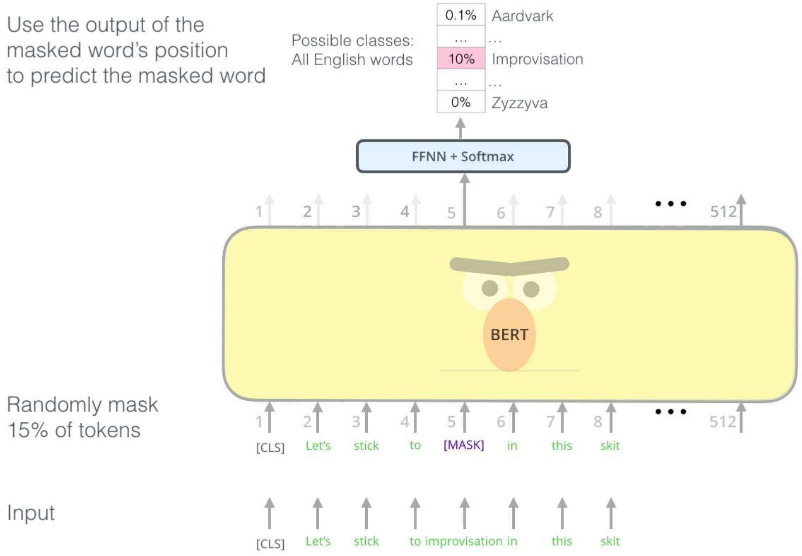


(d) RNN based sequential recommendation methods.

INTRODUCTION

Bidirectional architectures:

all items in the bidirectional model can leverage the contexts from both left and right side.



[The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)]

INTRODUCTION

Jointly conditioning on both left and right context in a deep bidirectional model would cause information leakage

Cloze task (Masked Language Model)

Every now and then I feel ___ a newly-qualified teacher who wants ___ try everything new that he comes ___ (honestly it doesn't happen that often any more ___ when it does I take advantage ___ it!) ___ one ___ these moments, I tried this amazing piece ___ software ___ create cloze texts

RELATED WORK

Sequential Recommendation:

- Markov chains (MCs)
- RNN
- Convolutional Sequence Model (Caser)

Attention Mechanism:

- RNN with attention mechanism
- Self-Attentive Sequential Recommendation (SASRec)

Problem Statement

- $\mathcal{U}=\{u_1, u_2, \ldots, u_{|\mathcal{U}|}\}$- set of users
- $\mathcal{V}=\{v_1, v_2, \ldots, v_{|\mathcal{V}|}\}$- set of items
- $\mathcal{S}_u=[v_1^{(u)}, \ldots, v_t^{(u)}, \ldots, v_{n_u}^{(u)}]$ - interaction sequence in chronological order for user u ∈ U

It can be formalized as modeling the probability over all possible items for user u at time step n_u+1:

$$p(v_{n_u+1}^{(u)} = v | \mathcal{S}_u)$$

#ods_recommender_systems

BERT4Rec Architecture

Position-wise Feed-Forward Network:

$$\mathrm{PFFN}(\boldsymbol{H}^l) = \left[\mathrm{FFN}(\boldsymbol{h}_1^l)^\top; \ldots; \mathrm{FFN}(\boldsymbol{h}_t^l)^\top\right]^\top$$

$$\mathrm{FFN}(\boldsymbol{x}) = \mathrm{GELU}\left(\boldsymbol{x}\boldsymbol{W}^{(1)} + \boldsymbol{b}^{(1)}\right)\boldsymbol{W}^{(2)} + \boldsymbol{b}^{(2)} \quad (3)$$

$$\mathrm{GELU}(x) = x\Phi(x)$$

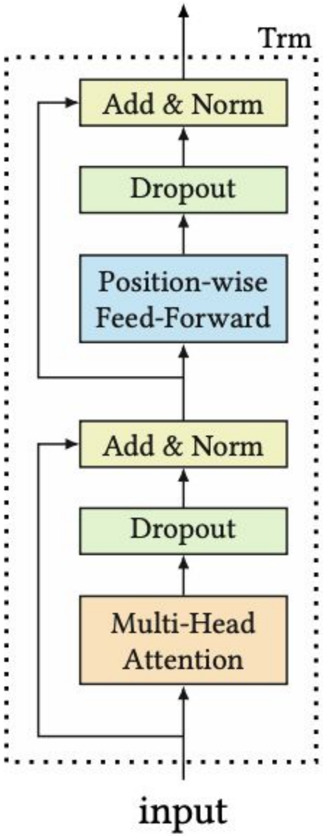
LN(x + Dropout(sublayer(x)))

Multi-Head Self-Attention:

$$\mathrm{MH}(\boldsymbol{H}^l) = [\mathrm{head}_1; \mathrm{head}_2; \ldots; \mathrm{head}_h]\boldsymbol{W}^O$$

$$\mathrm{head}_i = \mathrm{Attention}\left(\boldsymbol{H}^l\boldsymbol{W}_i^Q, \boldsymbol{H}^l\boldsymbol{W}_i^K, \boldsymbol{H}^l\boldsymbol{W}_i^V\right) \quad (1)$$

$$\mathrm{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d/h}}\right)\boldsymbol{V} \quad (2)$$



(a) Transformer Layer.

BERT4Rec Architecture

Output Layer:

$$P(v) = \mathrm{softmax}\left(\mathrm{GELU}(\boldsymbol{h}_t^L \boldsymbol{W}^P + \boldsymbol{b}^P)\boldsymbol{E}^\top + \boldsymbol{b}^O\right)$$

Stacking Transformer Layer:

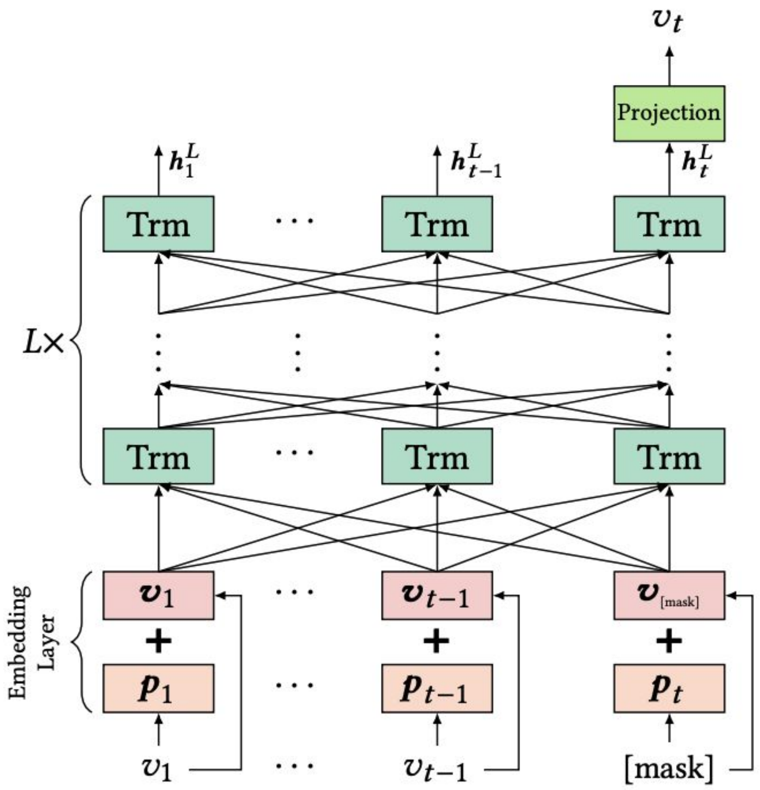
$$\boldsymbol{H}^l = \mathrm{Trm}(\boldsymbol{H}^{l-1}), \quad \forall i \in [1, \ldots, L] \tag{4}$$

$$\mathrm{Trm}(\boldsymbol{H}^{l-1}) = \mathrm{LN}\Big(\boldsymbol{A}^{l-1} + \mathrm{Dropout}\big(\mathrm{PFFN}(\boldsymbol{A}^{l-1})\big)\Big) \tag{5}$$

$$\boldsymbol{A}^{l-1} = \mathrm{LN}\Big(\boldsymbol{H}^{l-1} + \mathrm{Dropout}\big(\mathrm{MH}(\boldsymbol{H}^{l-1})\big)\Big) \tag{6}$$

Embedding Layer:

$$\boldsymbol{h}_i^0 = \boldsymbol{v}_i + \boldsymbol{p}_i$$



(b) BERT4Rec model architecture.

#ods_recommender_systems

Model Learning

Training:

Input: $[v_1, v_2, v_3, v_4, v_5] \xrightarrow{\text{randomly mask}} [v_1, [\mathtt{mask}]_1, v_3, [\mathtt{mask}]_2, v_5]$

Labels: $[\mathtt{mask}]_1 = v_2, \quad [\mathtt{mask}]_2 = v_4$

Negative log-likelihood loss

$$\mathcal{L} = \frac{1}{|\mathcal{S}_u^m|} \sum_{v_m \in \mathcal{S}_u^m} -\log P(v_m = v_m^* | \mathcal{S}_u') \tag{8}$$

Test:

1. Cloze objective is to predict the current masked item
2. Sequential recommendation aims to predict the future

1. Append the special token "[mask]" to the end of user's behavior sequence
2. Produce samples that only mask the last item in the input sequences during training

EXPERIMENT

Table 1: Statistics of datasets.

Datasets	#users	#items	#actions	Avg. length	Density
Beauty	40,226	54,542	0.35m	8.8	0.02%
Steam	281,428	13,044	3.5m	12.4	0.10%
ML-1m	6040	3416	1.0m	163.5	4.79%
ML-20m	138,493	26,744	20m	144.4	0.54%

EXPERIMENT

Table 2: Performance comparison of different methods on next-item prediction. Bold scores are the best in each row, while underlined scores are the second best. Improvements over baselines are statistically significant with $p < 0.01$.

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%

EXPERIMENT

Question 1: Do the gains come from the bidirectional self-attention model or from the Cloze objective?

Table 3: Analysis on bidirection and Cloze with $d = 256$.

Model	Beauty			ML-1m		
	HR@10	NDCG@10	MRR	HR@10	NDCG@10	MRR
SASRec	0.2653	0.1633	0.1536	0.6629	0.4368	0.3790
BERT4Rec (1 mask)	0.2940	0.1769	0.1618	0.6869	0.4696	0.4127
BERT4Rec	0.3025	0.1862	0.1701	0.6970	0.4818	0.4254

Question 2: Why and how does bidirectional model outperform unidirectional models?

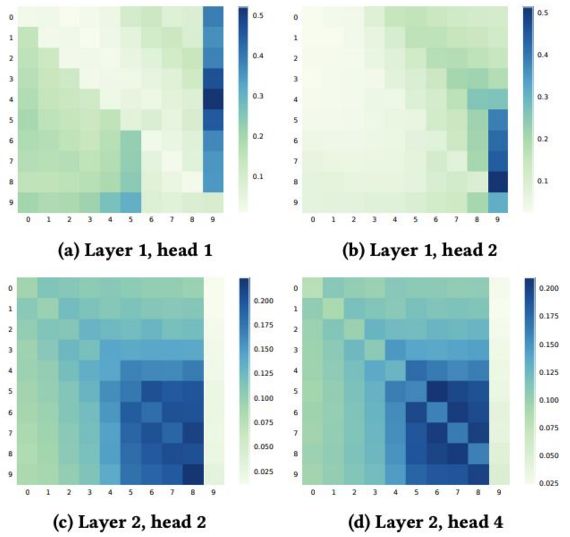


Figure 2: Heat-maps of average attention weights on Beauty, the last position "9" denotes "[mask]" (best viewed in color).

#ods_recommender_systems

EXPERIMENT

Impact of Hidden Dimensionality d

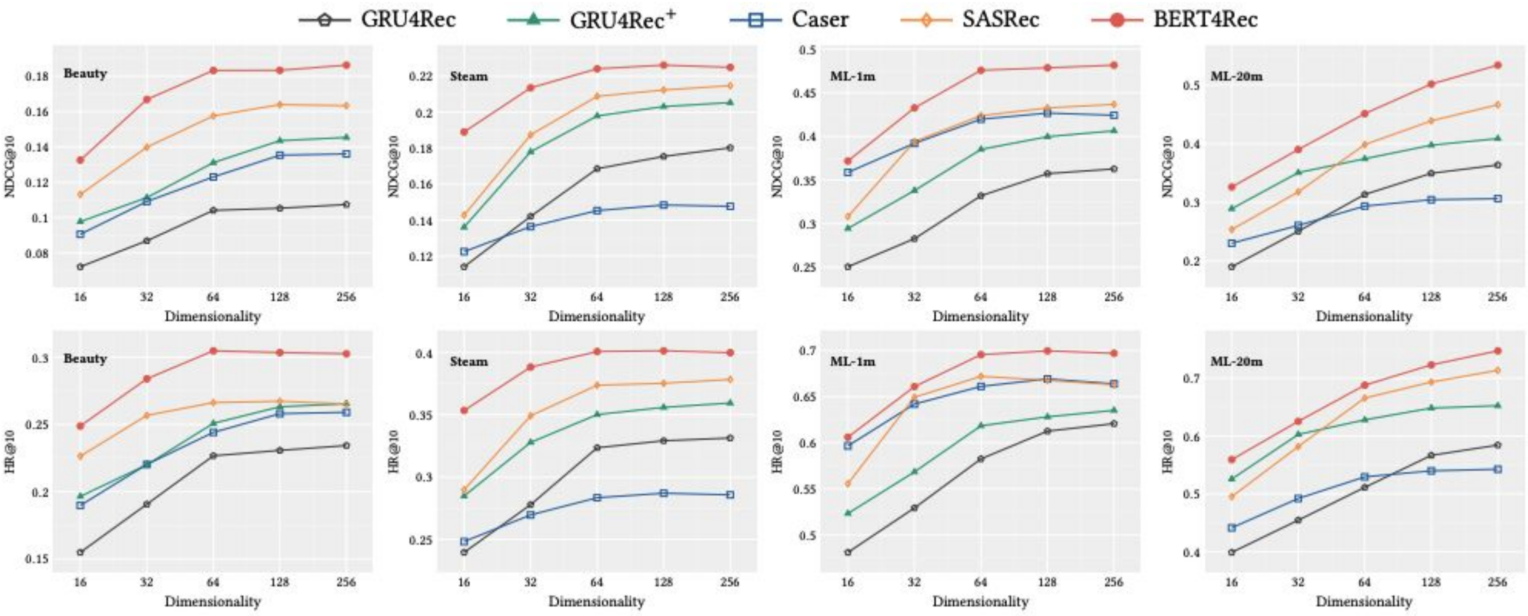


Figure 3: Effect of the hidden dimensionality d on HR@10 and NDCG@10 for neural sequential models.

#ods_recommender_systems

EXPERIMENT

Impact of Mask Proportion ρ

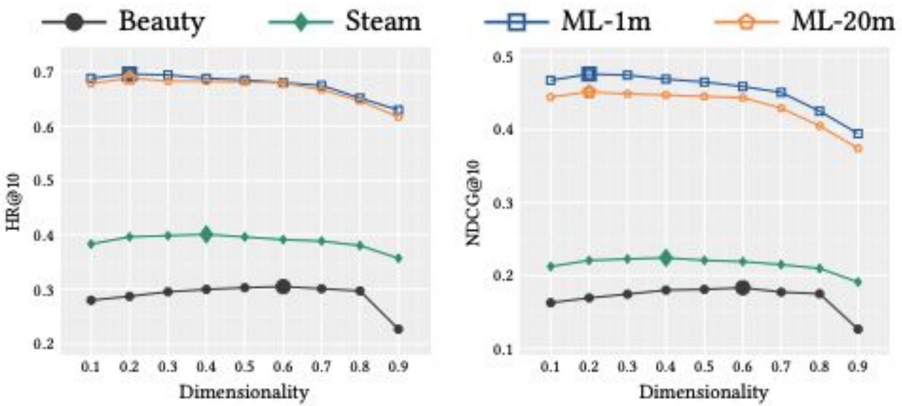


Figure 4: Performance with different mask proportion ρ on $d = 64$. Bold symbols denote the best scores in each line.

EXPERIMENT

Impact of Maximum Sequence Length N

Table 4: Performance with different maximum length N.

		10	20	30	40	50
Beauty	#samples/s	5504	3256	2284	1776	1441
	HR@10	0.3006	0.3061	0.3057	0.3054	0.3047
	NDCG@10	0.1826	0.1875	0.1837	0.1833	0.1832

		10	50	100	200	400
ML-1m	#samples/s	14255	8890	5711	2918	1213
	HR@10	0.6788	0.6854	0.6947	0.6955	0.6898
	NDCG@10	0.4631	0.4743	0.4758	0.4759	0.4715

Ablation Study

Table 5: Ablation analysis (NDCG@10) on four datasets. Bold score indicates performance better than the default version, while ↓ indicates performance drop more than 10%.

Architecture	Dataset			
	Beauty	Steam	ML-1m	ML-20m
$L = 2, h = 2$	0.1832	0.2241	0.4759	0.4513
w/o PE	0.1741	0.2060	0.2155↓	0.2867↓
w/o PFFN	0.1803	0.2137	0.4544	0.4296
w/o LN	0.1642↓	0.2058	0.4334	0.4186
w/o RC	0.1619↓	0.2193	0.4643	0.4483
w/o Dropout	0.1658	0.2185	0.4553	0.4471
1 layer $(L = 1)$	0.1782	0.2122	0.4412	0.4238
3 layers $(L = 3)$	**0.1859**	**0.2262**	**0.4864**	**0.4661**
4 layers $(L = 4)$	**0.1834**	**0.2279**	**0.4898**	**0.4732**
1 head $(h = 1)$	**0.1853**	0.2187	0.4568	0.4402
4 heads $(h = 4)$	0.1830	**0.2245**	**0.4770**	**0.4520**
8 heads $(h = 8)$	0.1823	**0.2248**	0.4743	**0.4550**

#ods_recommender_systems

Links

1. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer
2. Self-Attentive Sequential Recommendation
3. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
4. The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)
5. Author's implementation on tensorflow
6. A Systematic Review and Replicability Study of BERT4Rec for Sequential Recommendation