CARCA: Context and Attribute-Aware Next-Item Recommendation via Cross-Attention

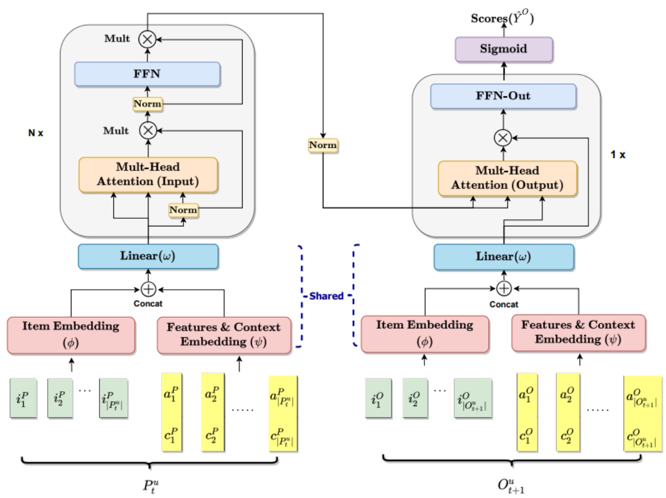

Figure 1: Illustration of the CARCA model, which is composed of two main branches, namely the profile-level features extraction branch on the left and the target items cross-attention scoring branch on the right.

Problem Definition

An item recommendation problem consists of a set $\mathcal{U} \coloneqq \{1, \dots, U\}$ of **users**, a set $\mathcal{I} \coloneqq \{1, \dots, I\}$ of **items** and a sequence $\mathcal{D} = \big((u_1, i_1), \dots, (u_N, i_N)\big) \in (\mathcal{U} \times \mathcal{I})^*$ of their past **interactions** originating from an unknown distribution p on user/item pairs (with $U, I, N \in \mathbb{N}$). Sought is a model $\hat{p}: \mathcal{U} \to (\mathbb{R}_0^+)^I$ for the unknown conditional density $p(i \mid u)$, i.e., given a loss function $\mathcal{L}: \mathcal{I} \times (\mathbb{R}_0^+)^I \to \mathbb{R}$ with minimal expected loss

$$\mathbb{E}_{(u,i)\sim p}\ \mathcal{L}(i, \hat{p}(u))$$

One calls the problem **having item attributes**, if additionally there is a given matrix $A^{IT} \in \mathbb{R}^{I \times j}$ containing for item an attribute vector with $j \in \mathbb{N}$ attribute values each.

One calls the problem **having context**, if each interaction has additional attributes, i.e., $\mathcal{D} = \big((u_1, i_1, c_1), \dots, (u_N, i_N, c_N)\big) \in \big(\mathcal{U} \times \mathcal{I} \times \mathbb{R}^l\big)^*$ (with $l \in \mathbb{N}$) is a sample from an unknown distribution on user/item/context triples and the goal is to find a model $\hat{p}: \mathcal{U} \times \mathbb{R}^l \to (\mathbb{R}_0^+)^I$ for the unknown conditional density $p(i \mid u, c)$, i.e., given a loss function $\mathcal{L}$ with minimal expected loss

$$\mathbb{E}_{(u,i,c)\sim p}\ \mathcal{L}(i, \hat{p}(u, c))$$

The most frequently encountered context is an absolute **time-stamp** at which the user interacted with an item (for example measured as a real number in Unix Time).

Sequential approaches usually consider all users to have profiles P_t^u that contain the sequence of their previously interacted items $P_t^u \coloneqq \{i_1^P, \ldots, i_{|P_t^u|}^P\}$ along with their attributes $A_t^u \in \mathbb{R}^{|P_t^u| \times j}$ and their interactions' contextual features $C_t^u \in \mathbb{R}^{|P_t^u| \times l}$ such as timestamps. The main goal of the sequential item recommendation task will be to rank a target list of items $O_{t+1}^u \coloneqq \{i_1^O, \ldots, i_{|O_{t+1}^u|}^O\}$ based on their likelihood of being interacted with by the target user u at time $t+1$ while similarly considering their attributes and contextual features existing at that time point.

Model Design

Embedding

We utilize two separate dedicated embedding functions ϕ and ψ. The first embedding function $\phi: \mathbb{R}^I \rightarrow \mathbb{R}^d$ is used to extract the first half of the item's latent features $z_i \in \mathbb{R}^d, i \in P_t^u \cup O_{t+1}^u$ from the item's one-hot encoded vectors $x_i \in \mathbb{R}^I$. The second function $\psi: \mathbb{R}^{j+l} \rightarrow \mathbb{R}^g$ extracts the second half of the latent features $q_i \in \mathbb{R}^g$ from the item's contextual features $c_i \in \mathbb{R}^l$ and attributes $a_i \in \mathbb{R}^j$. After extracting the two partial latent feature vectors, both of them are concatenated and fed into a third embedding layer $\omega: \mathbb{R}^{g+d} \rightarrow \mathbb{R}^d$ to generate the final item's latent features $e_i \in \mathbb{R}^d$ as follows:

$$z_i = \phi(x_i) = x_i W^\phi + b^\phi, \;\; W^\phi \in \mathbb{R}^{I \times d}, \; b^\phi \in \mathbb{R}^d \tag{1}$$

$$q_i = \psi(a_i, c_i) = \text{concat}_{col}(a_i, c_i) W^\psi + b^\psi, \;\; W^\psi \in \mathbb{R}^{(j+l) \times g}, \; b^\psi \in \mathbb{R}^g \tag{2}$$

$$e_i = \omega(z_i, q_i) = \text{concat}_{col}(z_i, q_i) W^\omega + b^\omega, W^\omega \in \mathbb{R}^{(g+d) \times d}, b^\omega \in \mathbb{R}^d \tag{3}$$

The embedding pipeline is shared between user profile P_t^u and target items O_{t+1}^u.

Self-Attention Blocks

#TODO

Sampling and Loss Function

For each user we exclude his last interaction and we convert the user profile sequence into a fixed-length input list of items $P^u \coloneqq \{i_1^P, \ldots, i_{|P_t^u|-1}^P\}$ via truncation or padding.

The list of target items is constructed by combining a list of positive items $O^{u(+)}$ and another list of negative items $O^{u(-)}$ with equal length. The positive items list is constructed by right shifting the input list P^u to include the user's last interaction $O^{u(+)} \coloneqq \{i_2^P, \ldots, i_{|P_t^u|}^P\}$ while the negative items list is generated by selecting random negative items $i \notin P^u$ and they are given the same contextual features as their corresponding positive ones.

Finally, we optimize the CARCA model by minimizing the binary cross-entropy loss using an ADAM optimizer, and the padded items are masked to prevent them from contributing to the loss function.

$$\mathcal{L} = -\sum_{u\in U}\sum_{r\in O^{u(+)}\cup O^{u(-)}} \left(Y_r^O \log(\hat{Y}_r^O) + (1-Y_r^O)\log(1-\hat{Y}_r^O)\right) \tag{11}$$

Experiments

Table 2: Performance comparison of the CARCA against state-of-the-art sequential (SEQ), context (CXT) and attribute-aware (ATT) recommendation models.

Model	ATT	CXT	SEQ	Men HR@10	Men NDCG@10	Fashion HR@10	Fashion NDCG@10	Games HR@10	Games NDCG@10	Beauty HR@10	Beauty NDCG@10
Random				0.098	0.044	0.099	0.045	0.100	0.045	0.099	0.045
TopPop				0.415	0.269	0.407	0.262	0.519	0.314	0.451	0.261
EASE [22]				0.193	0.133	0.213	0.146	0.623	0.465	0.299	0.222
GraphRec [16]	✓			0.374	0.219	0.419	0.244	0.613	0.400	0.435	0.273
DeepFM [6]	✓	✓		0.334	0.237	0.283	0.185	0.736	0.494	0.464	0.266
SASRec [12]			✓	0.397	0.259	0.381	0.245	0.742	0.541	0.485	0.322
OAR [26]		✓	✓	0.355	0.225	0.340	0.214	0.704	0.496	0.485	0.329
TiSASRec [13]		✓	✓	0.333	0.194	0.384	0.234	0.748	0.533	0.492	0.333
BERT4Rec [23]			✓	0.315	0.193	0.328	0.209	0.705	0.509	0.478	0.318
SSE-SASRec [27]			✓	0.397	0.257	0.385	0.248	0.754	0.549	0.481	0.330
SSE-PT [28]			✓	0.397	0.258	0.381	0.246	0.748(0.775)	0.545(0.566)	0.443(0.502)	0.302(0.337)
S^3Rec [36]	✓		✓	0.365	0.238	0.367	0.239	0.765	0.549	0.538	0.371
SASRec++ (Our extension)	✓	✓	✓	0.500	0.315	0.546	0.344	0.752	0.533	0.545	0.351
CARCA (w/o CA) (Ours)	✓	✓	✓	0.521	0.322	0.568	0.359	0.738	0.517	0.556	0.358
CARCA (Ours)	✓	✓	✓	**0.550***	**0.349***	**0.591***	**0.381***	**0.782***	**0.573***	**0.579***	**0.396**
Improv. vs best published baseline (%)				38.65	35.87	53.71	53.24	2.20	4.38	7.70	6.74
Improv. vs SASRec++ (%)				10.09	10.79	8.25	10.67	3.96	7.64	6.31	12.95

(*) Significantly outperforms the best baseline at the 0.01 levels.
Published results of SSE-PT are indicated in parentheses.

Ablation Study

Table 4: Ablation analysis between different CARCA configurations on the Men dataset.

Configuration	HR@10	NDCG@10
Default (1)	**0.550**	**0.349**
Additive residual connections (2)	0.513	0.325
Concat. all features (3)	0.543	0.340
Concat. item features (4)	0.540	0.339
Positional encoding (5)	0.544	0.345
Additional self-attention blocks on output (6)	0.427	0.231
CARCA with single target split (7)	0.394	0.233
CARCA with transformer architecture (8)	0.459	0.276

Runtime Comparison

Table 6: Runtime Comparison on Games Dataset

Model	Average batch runtime in seconds
S^3Rec	0.580
SSE-PT	0.008
SASRec	0.013
TiSASRec	0.075
SSE-SASRec	0.015
OAR	0.018
SASRec++	0.028
CARCA (w/o CA)	0.015
CARCA	0.026