Attention-Based Transactional Context Embedding for Next-Item Recommendation

2021.06.07.

박충현

Contents

- **Transaction-Based Recommender Systems**
- **Contribution**
- **Model Architecture**
- **Training**
  - Objective, algorithm
- **Experiment**

Transaction-Based Recommender Systems

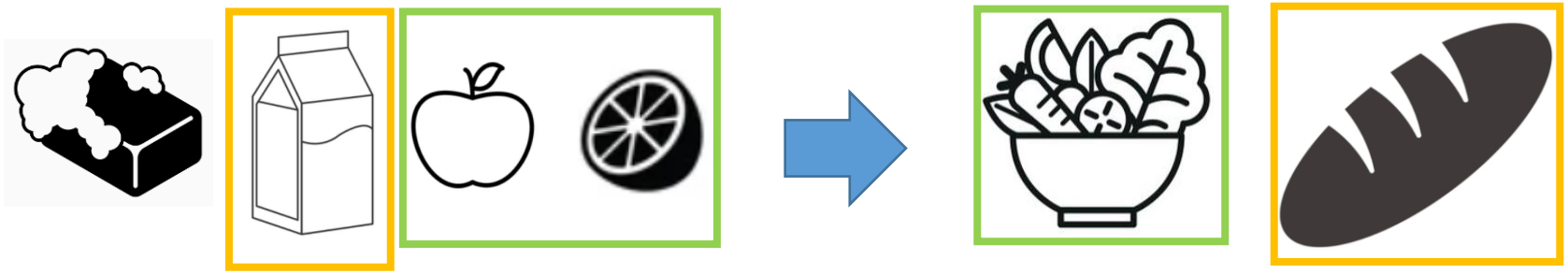
- **RS issue: Repeat items similar to the items that users already chose**

- **Solution: Recommend the next item to a user in a transactional context**
  - Transactional context: items observable in a transaction
    - Ex) shopping-basket record
  - Should not repeat items that users have already chosen

TID	Items in the Basket
1	espresso, sugar, newspaper
2	espresso, sugar, cola
3	espresso, sugar
4	cappuccino, cigarettes
5	cappuccino, sugar
6	cappuccino, sugar, sweets
7	decaf, sugar, chewing_gums
8	decaf, soda, vinegar
9	decaf, sugar, cigarettes

Transaction-Based Recommender Systems

- **Challenge**
  - Ex) Current transaction = {soap, milk, apple, orange}



  - No rigid order
  - Irrelevant items exist for each next-item recommendation

- **Need to learn relevance and transition between items in a transactional context**

Contribution

- **Existing methods**
  - MC, MF, RNN, …
  - Designed for time-series data with rigid order
  - Not paying attention to relevant items

- **ATEM**
  - Attention-based Transaction Embedding Model
  - Attentive context embedding
    - Intensify relevant item, weaken irrelevant item to the next choice
    - No rigid ordering assumption
  - Effective, efficient network on a large number of items
  - SOTA in accuracy, novelty

Problem Statement

- **Set of all transactions $T = \{t_1, t_2, \ldots, t_{|T|}\}$**
- **Whole item set $I = \{i_1, i_2, \ldots, i_{|I|}\}$**
  - No rigid order assumption
- **Transaction $t = \{i_1, i_2, \ldots, i_{|t|}\}$**

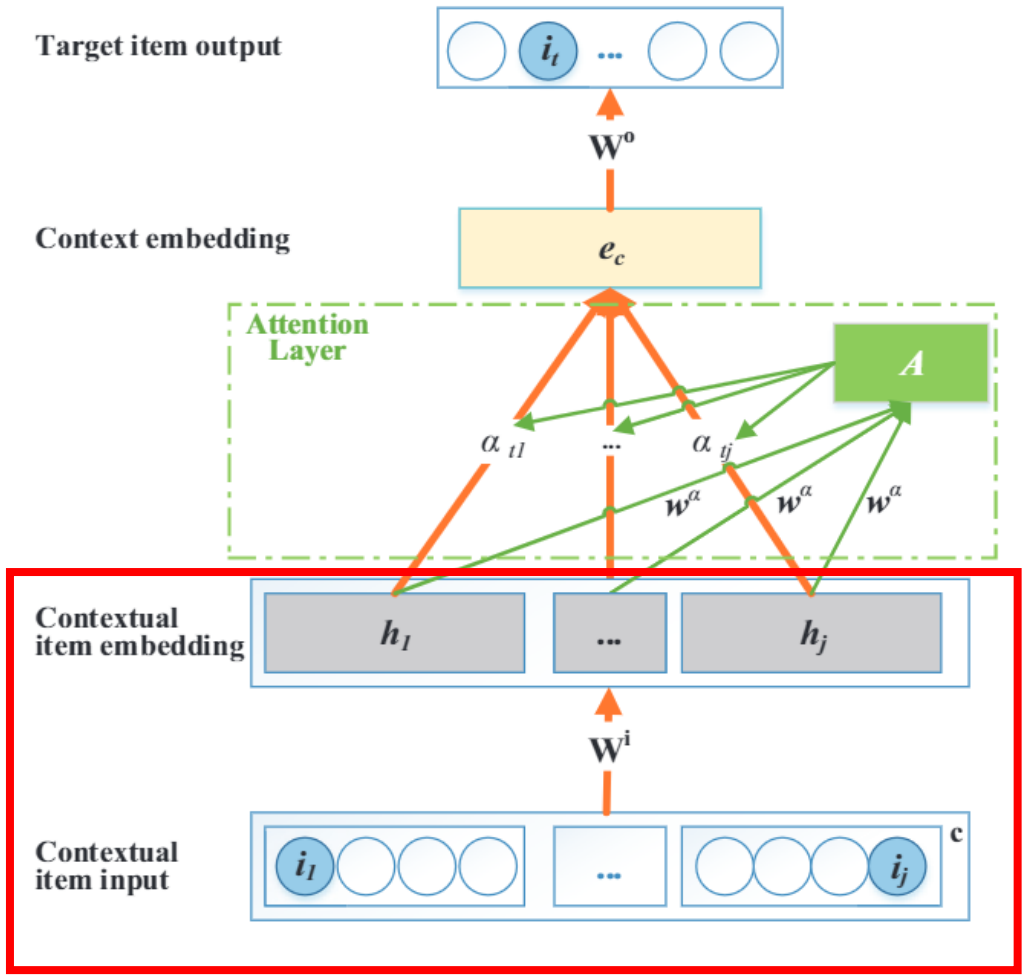
- **Training ATEM: predict $P(i_s|c)$ for each item in a transaction**
  - Context c = all items in t except i_s ($i_s \in t$)
  - Target item i_s : each item in the given transaction
  - $|t|$ training instances per a transaction

Model Architecture

- **Contextual item input**
  - One-hot encoding for each item
  - Contextual itemset c as input

- **Contextual item embedding**
  - Sparse one-hot vector into informative dense vector
  - Learn input weight matrix W^i
  - Get embedding of jth item h_j

$$\mathbf{h}_j = \mathbf{W}^i_{:,j}$$



Model Architecture

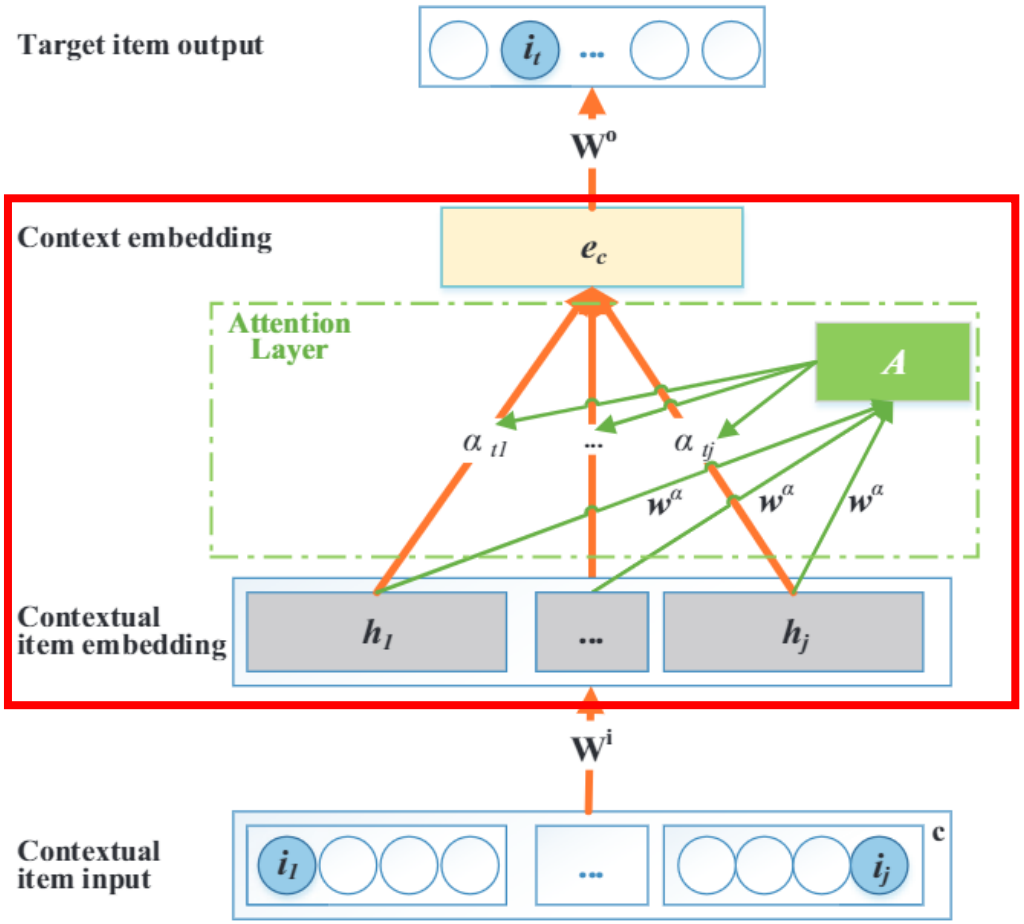
- **Context embedding e_c of context c**
  - Integrate all item embeddings in c
  - α_{tj}= contribution scale of i_j to the occurrence of i_t

$$\mathbf{e}_c = \sum_{i_j \in c} \alpha_{tj} \mathbf{h}_j, \quad s.t. \sum_{i_j \in c} \alpha_{tj} = 1$$

- **Weight of contextual items**
  - Softmax on transformed item embeddings
  - w^{α}= context vector shared by all contextual items
  - "What is the informative item?"

$$\alpha_{tj} = \frac{exp(e(\mathbf{h}_j))}{\sum_{s \in \mathbf{c}_t} exp(e(\mathbf{h}_s))}$$

$$e(\mathbf{h}_j) = \mathbf{w}^{\alpha} \mathbf{h}_j^T$$



Model Architecture

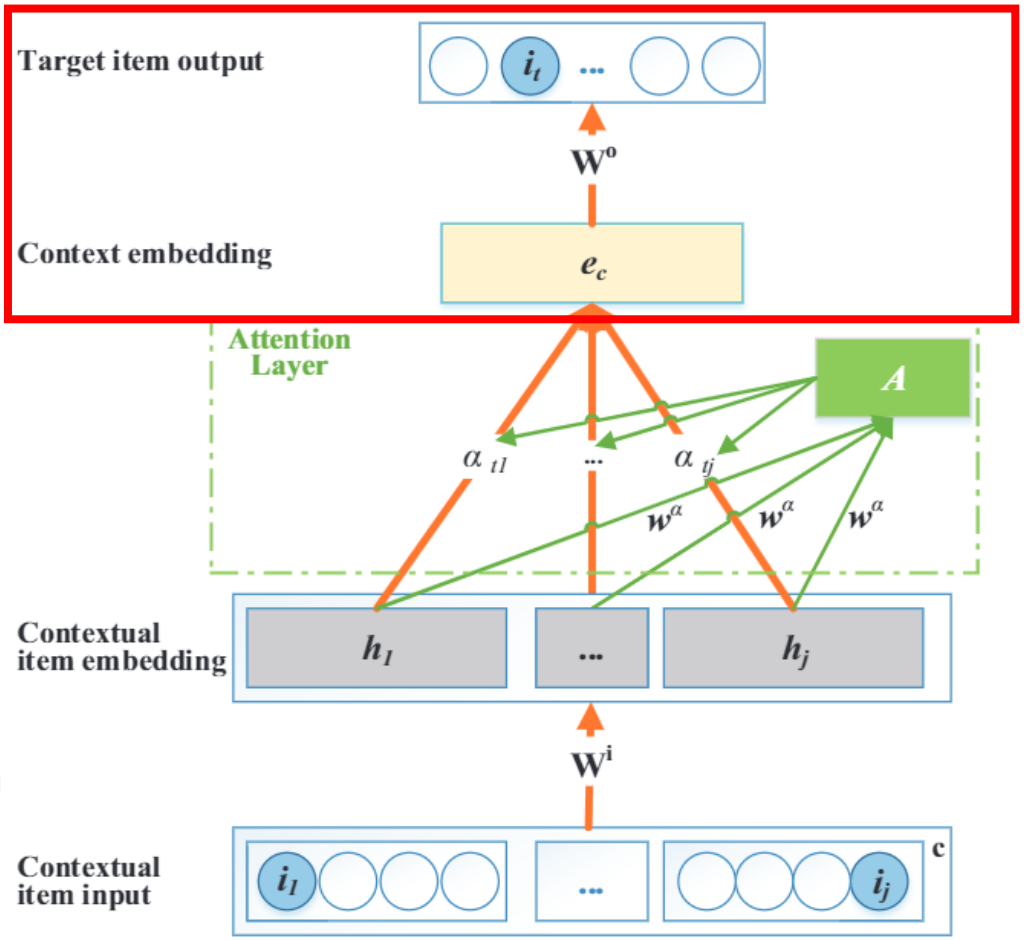
- **Score of each item**
  - Learn output weight matrix W^o
  - Fully connected layer
  - $S_t(c)$ = score of a target item i_t w.r.t. context c

$$S_t(\mathbf{c}) = \mathbf{W}^o_{t,:}\mathbf{e_c}$$

- **Target item prediction**
  - Softmax

$$P_\Theta(i_t|\mathbf{c}) = \frac{exp(S_t(\mathbf{c}))}{Z(\mathbf{c})}$$

$$Z(\mathbf{c}) = \sum_{i \in I} exp(S_i(\mathbf{c}))$$



Training Objective

- **Training dataset D={<c,i_c>}**
  - Input c: context, output i_c: observed output

- **Training objective**
  - Learning θ to maximize conditional log-likelihood

$$\Theta = \{\mathbf{W}^i, \mathbf{w}^\alpha, \mathbf{W}^o\}$$

$$L_\Theta = \sum_{d \in D} logP_\Theta(i_c|\mathbf{c}) = \sum_{d \in D} S_{i_c}(\mathbf{c}) - logZ(\mathbf{c})$$

  - Too much time spent on Z(c)

$$Z(\mathbf{c}) = \sum_{i \in I} exp(S_i(\mathbf{c}))$$

- **Use different objective with similar effect**
  - Noise Contrastive Estimation

Noise Contrastive Estimation

- **Noise Contrastive Estimation (NCE)**
  - Positive sample from data distribution, noises from known noise distribution Q
  - Distinguish positive sample from noises
  - Approximate, cheaper computation

$$P_\Theta(y, i_c|\mathbf{c}) = \frac{1}{K+1} P_\Theta(i_c|\mathbf{c}) + \frac{K}{K+1} Q(i_c)$$

- **Probability calculation**
  - i_c is from true data distribution (positive example)

$$P_\Theta(y = 1|i_c, \mathbf{c}) = \frac{P_\Theta(i_c|\mathbf{c})}{P_\Theta(i_c|\mathbf{c}) + KQ(i_c)} \approx \frac{exp(S_{i_c}(\mathbf{c}))}{exp(S_{i_c}(\mathbf{c})) + KQ(i_c)}$$

  - i_c is from noise distribution

$$P_\Theta(y = 0|i_c, \mathbf{c}) = 1 - P_\Theta(y = 1|i_c, \mathbf{c})$$

Noise Contrastive Estimation

- **Goal: maximize likelihood of truth against K noise samples**
  - No Z(c), reduced computation cost

$$J_\Theta(i_c, \mathbf{c}) = logP_\Theta(y=1|i_c, \mathbf{c}) + K\mathbb{E}_{i_k \sim Q}[logP_\Theta(y=0|i_c, \mathbf{c})]$$

$$\approx logP_\Theta(y=1|i_c, \mathbf{c}) + \sum^{K} logP_\Theta(y=0|i_k, \mathbf{c})$$

  - Gradient approaches to that of original method as K increases
  - Apply on back-propagation

$$\nabla J_\Theta(i_c, \mathbf{c}) = \frac{KQ(i_c)}{exp(S_{i_c}(\mathbf{c})) + KQ(i_c)} \nabla S_{i_c}(\mathbf{c})$$

$$- \sum_{k=1}^{K} \frac{exp(S_{i_k}(\mathbf{c}))}{exp(S_{i_k}(\mathbf{c})) + KQ(i_k)} \nabla S_{i_k}(\mathbf{c})$$

Training Algorithm

- **Gradient based update on each parameter**

Algorithm 1 ATEM Parameter Learning Using SGD

$l \leftarrow 0$

while not converged **do**

  Compute output weight $w^o_{t,:}$-gradient (Eq. (5)): $g_{w^o_{t,:}} \leftarrow \mathbf{e}_c$

  Compute attention weight w^α_{tj}-gradient (Eq. (2-5)): $g_{w^\alpha_{tj}} \leftarrow \mathbf{W}^o_{t,:} \odot \mathbf{h}_j^2 \odot \nabla_{e(\mathbf{h}_j)} \alpha_{tj}$

  Compute input weight $w^i_{:,j}$-gradient (Eq. (1-5)): $g_{w^i_{:,j}} \leftarrow \mathbf{W}^{o\top}_{t,:} \odot (\alpha_{tj} + \nabla_{e(\mathbf{h}_j)} \alpha_{tj} \odot \mathbf{w}^\alpha \odot \mathbf{h}_j)$

  Perform SGD-updates for $w^o_{t,:}$, w^α_{tj} and $w^i_{:,j}$:

  $w^o_{t,:} \leftarrow w^o_{t,:} + S^l_t(g) g_{w^o_{t,:}}$ (output weight update),

  $w^\alpha_{tj} \leftarrow w^\alpha_{tj} + S^l_{tj}(g) g_{w^\alpha_{tj}}$ (attention weight update),

  $w^i_{:,j} \leftarrow w^i_{:,j} + S^l_j(g) g_{w^i_{:,j}}$ (input weight update)

  $l \leftarrow l + 1$

end while

Experiment

- **Baselines**
  - PBRS: Recommend with mined frequent patterns
  - FPMC: MF on the personalized transition matrix between items
  - PRME: Personalized ranking metric embedding, Markov chain
  - GRU4Rec: RNN-based session-based recommendation (paper 23)
  - TEM: In ATEM, replace attention mechanism with distance-based exponential decay
    - Larger weight on near items

- **Performance measure**
  - Accuracy + Novelty

Experiment

- **Performance metrics**
  - Accuracy: REC(Recall)@K, MRR
  - Novelty: MCAN@K
    - Non-overlap ratio between top-K recommendation and the corresponding context

$$MCAN = \frac{1}{N}\sum_{i=1}^{N}(1 - \frac{|R_i \cap \mathbf{c_i}|}{|R_i|})$$

- **Ex) N=1, top-2 recommendation**
  - c = {orange, milk, apple, green salad}
  - i_s= {bread}
  - R = {milk, butter}
  - MCAN@2 = 1 – ½ = 0.5

Experiment - Accuracy

- **Accuracy: best for all cases**
  - ATEM vs. TEM: Attention helps

Table 2: Accuracy comparisons on IJCAI-15

Model	REC@10	REC@50	MRR
PBRS	0.0780	0.0998	0.0245
FPMC	0.0211	0.0602	0.0232
PRME	0.0555	0.0612	0.0405
GRU4Rec	0.2283	0.3021	0.1586
ATEM	**0.3542**	**0.5134**	**0.2041**
TEM	0.3177	0.3796	0.1918

Table 3: Accuracy comparisons on Tafang

Model	REC@10	REC@50	MRR
PBRS	0.0307	0.0307	0.0133
FPMC	0.0191	0.0263	0.0190
PRME	0.0212	0.0305	0.0102
GRU4Rec	0.0628	0.0907	0.0271
ATEM	**0.1089**	**0.2016**	**0.0347**
TEM	0.0789	0.1716	0.0231

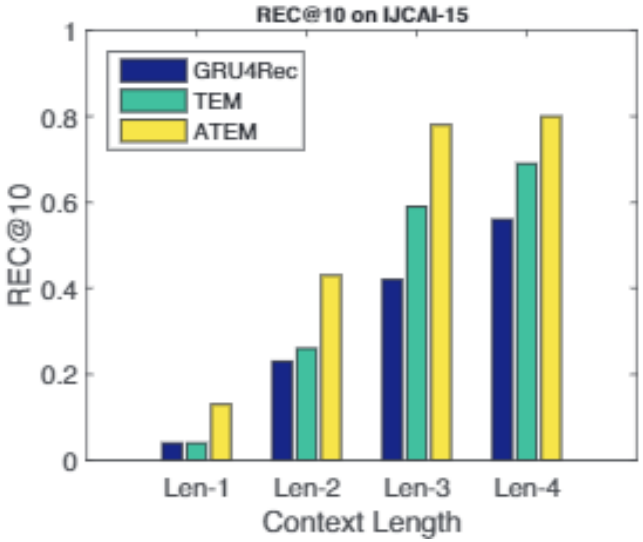
- **Accuracy(disordered)**
  - ATEM is the best
  - ATEM is better for disordered data

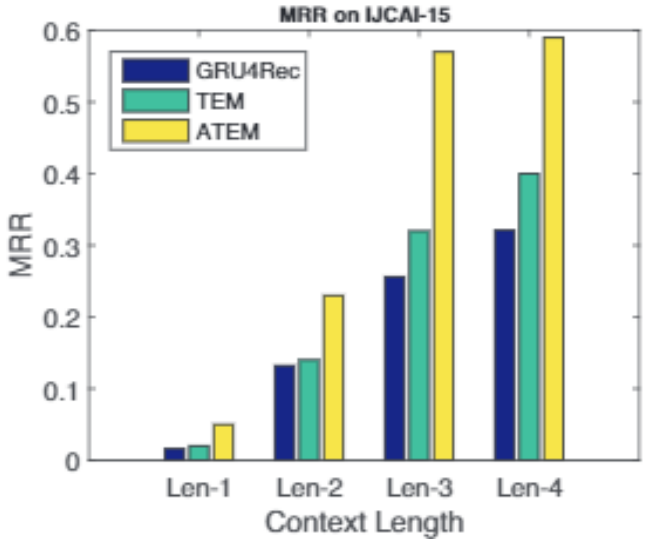
Table 4: Accuracy on disordered IJCAI-15

Model	REC@10	REC@50	MRR
PBRS	0.0500	0.0559	0.0185
FPMC	0.0151	0.0412	0.0183
PRME	0.0346	0.0389	0.0351
GRU4Rec	0.1636	0.2121	0.1022
ATEM	**0.3423**	**0.4981**	**0.1960**
TEM	0.2660	0.3012	0.1431

Experiment - Accuracy

- **Context length**
  - Longer context: more data, but fragile
  - ATEM outperforms other methods
  - PBRS, FPMC, PRME: not sensible to context length, not tested





Experiment - Novelty

- **Are novel (different) recommendations good?**
  - When high accuracy guaranteed, answers are also highly relevant

- **Highest novelty on ATEM**
  - Consider whole context + build attentive context embedding