Attention-Based Transactional Context Embedding for Next-Item Recommendation (AAAI 2018)

Jongjin Kim
Data Mining Lab
Dept. of CSE
Seoul National University

Overview

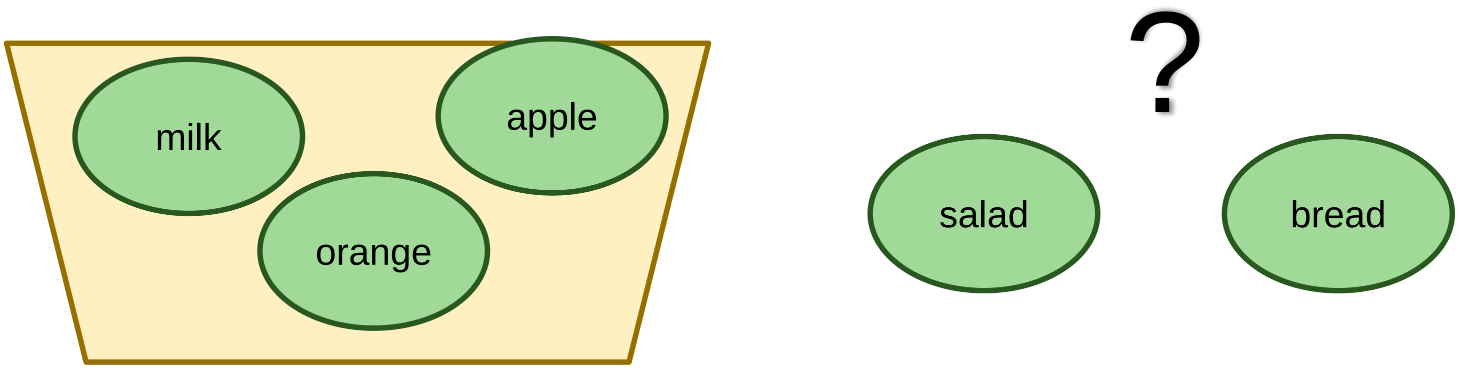
- RSs play an important role in real-world business.
- Users prefer items that are novel and different from that already in hand, but previous RSs tend to repeat items that are similar to already chosen.
- Transaction based RSs are quite different from traditional RSs built on user preferences and item properties.
- Attention-Based Transaction Approach can solve above issue.

Outline

- **Problem Definition**
- Proposed Method
- Experiments
- Conclusion

Context

- What is context?



- Traditional RSs would recommend salad since it is more similar with fruits.
- However this customer actually pick up bread to eat with milk while shopping.

Problem Definition

- Given
  - Set of items $I = \{i_1 \dots i_{|I|}\}$
  - Set of transactions $T = \{t_1 \dots t_{|T|}\}, (t = \{i_1 \dots i_{|t|}\})$
  - Context $c = t \backslash i_s$
- Generate
  - probability distribution $P(i_s|c)$

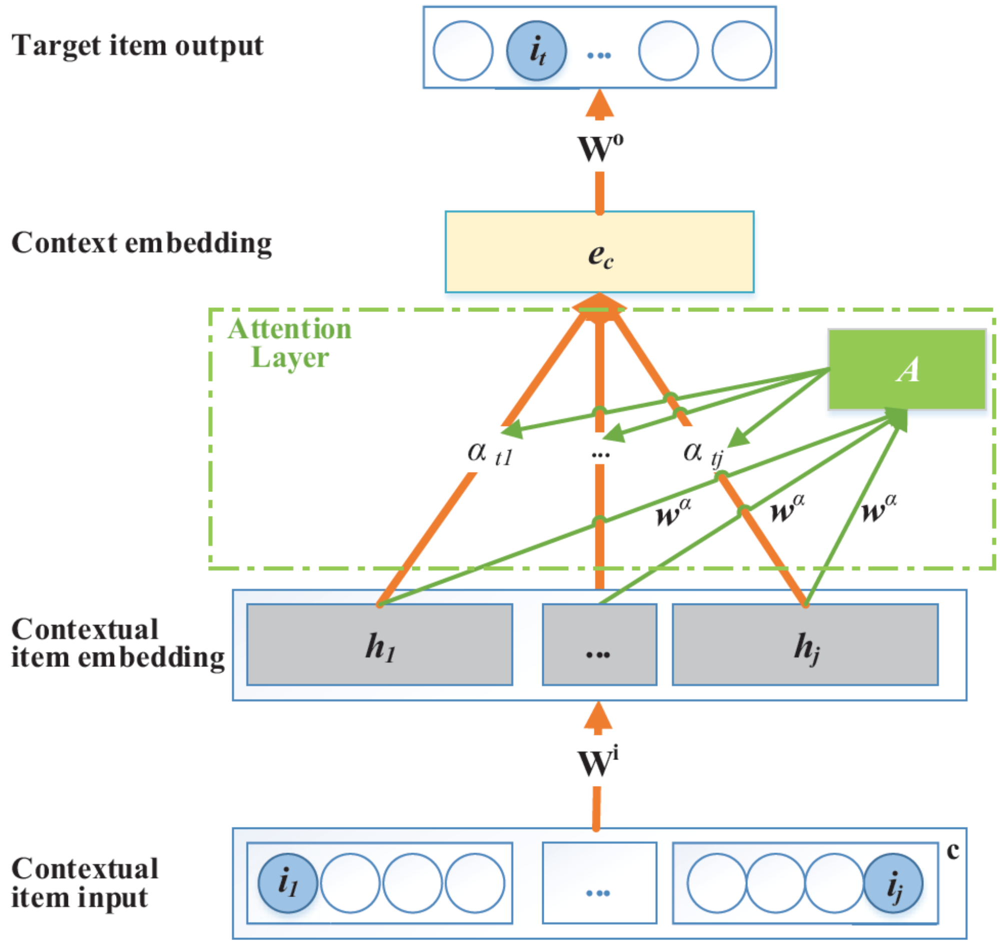
Challenges

- Context is hard to learn in previous approaches.
- Markov Chain
  - Only captures the first-order transition
- Matrix Factorization
  - Easily suffers from sparsity issues
- RNN
  - Too high computational cost for large data

Outline

- Problem Definition
- **Proposed Method**
- Experiments
- Conclusion

Proposed Method

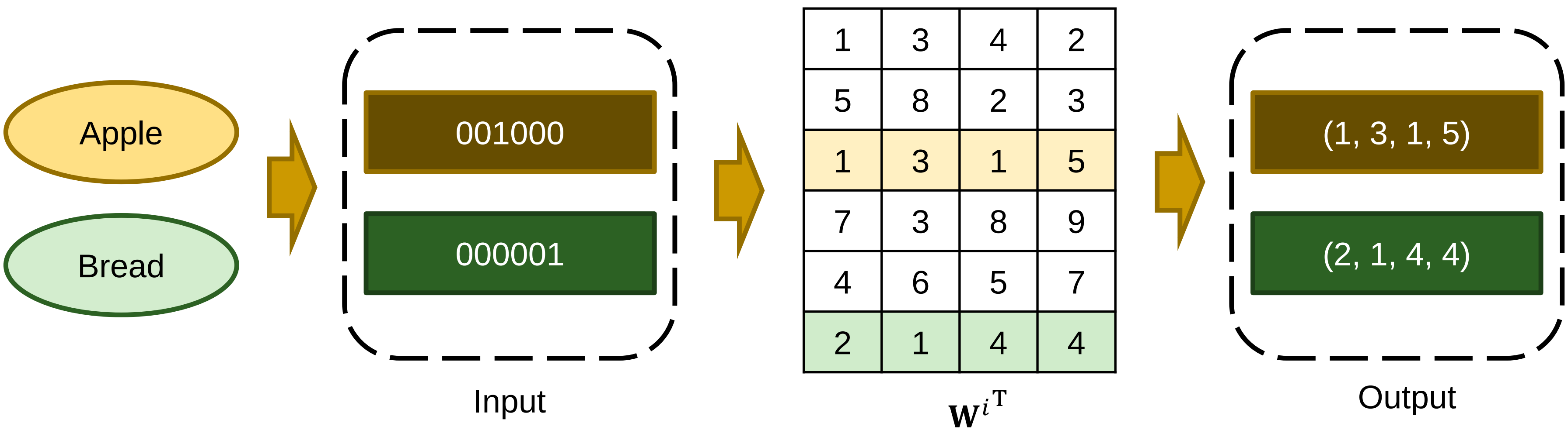


Input Layer

- Input: set of one-hot encoded items
- Output: set of contextual item embeddings

$$\mathbf{h}_j = \mathbf{W}^i_{:,j} \qquad \mathbf{W}^i \in \mathbb{R}^{K \times |I|}$$

- $\mathbf{W}^i$ performs as an embedding lookup table.

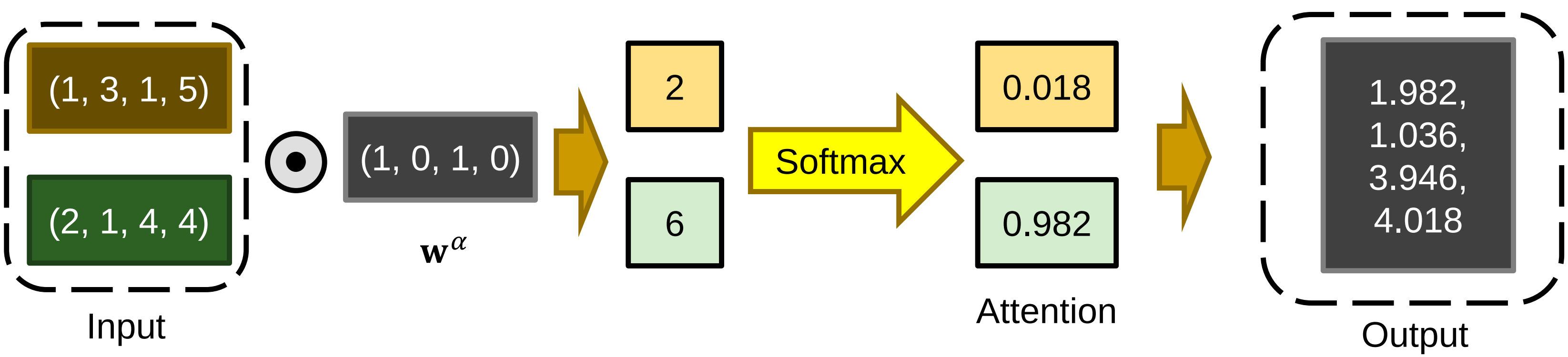


Context Layer (Attention)

- Input: set of contextual embeddings
- Output: context vector

$$\mathbf{e}_c = \sum_{i_j \in \mathbf{c}} \alpha_{tj} \mathbf{h}_j \qquad \alpha_{tj} = \frac{exp(e(\mathbf{h}_j))}{\sum_{s \in \mathbf{c}_t} exp(e(\mathbf{h}_s))} \qquad e(\mathbf{h}_j) = \mathbf{w}^{\alpha} \mathbf{h}_j^T$$

- Attention vector indicates which vector should be emphasize to describe context.



Output Layer

- Input: context vector
- Output: probability distribution over all items

$$\mathbf{W}^o \in \mathbb{R}^{|I|\times K} \qquad S_t(\mathbf{c}) = \mathbf{W}^o_{t,:}\mathbf{e}_{\mathbf{c}} \qquad P_\Theta(i_t|\mathbf{c}) = \frac{exp(S_t(\mathbf{c}))}{Z(\mathbf{c})} \qquad Z(\mathbf{c}) = \sum_{i\in I} exp(S_i(\mathbf{c}))$$

1.982,
1.036,
3.946,
4.018

Input

$\odot$

.4	.1	.8	.7
.3	.1	.1	.2
.2	.4	.4	.3
.2	.3	.1	.4
.5	.4	.7	.2
.7	.3	.1	.9

$\mathbf{W}^o$

6.866
1.896
3.595
2.709
4.971
5.709

$S_t(\mathbf{c})$

Softmax

0.6556
0.0046
0.0249
0.0103
0.0986
0.2062

Output

Rec

Learning

- Given dataset $D = \{\langle \mathbf{c}, i_c \rangle\}$
- joint probability distribution $P_\Theta(D) \propto \prod_{d \in D} P_\Theta(i_c|\mathbf{c})$
- => Maximize $L_\Theta = \sum_{d \in D} log P_\Theta(i_c|\mathbf{c}) = \sum_{d \in D} S_{i_c}(\mathbf{c}) - \boxed{log Z(\mathbf{c})}$

high computational cost

  - Use NCE(Noise Contrastive Estimation) to drop Z term.

Outline

- Problem Definition
- Proposed Method
- **Experiments**
- Conclusion

Experimental Setup

Table 1: Statistics of experimental datasets

Statistics	IJCAI-15	Tafang
#Transactions	144,936	19,538
#Items	27,863	5,263
Avg. Transaction Length	2.91	7.41
#Training Transactions	141,840	18,840
#Training Instances	412,679	141,768
#Testing Transactions	3,096	698
#Testing Instances	9,030	3,150

- test set: 20% of last 30 days' data
- evaluation metric: Recall@K, MRR

Comparison Methods

- PBRS
  - Typical pattern-based recommender system
- FPMC
  - Matrix factorization + first-order Markov chain
- PRME
  - Personalized ranking metric embedding method
  - Markov chain framework

Comparison Methods

- GRU4Rec
  - RNN-based approach
- TEM
  - ATEM without attention
  - Attention mechanism is replaced with distance-based exponential decay.

Results

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

Outline

- Problem Definition
- Proposed Method
- Experiments
- **Conclusion**

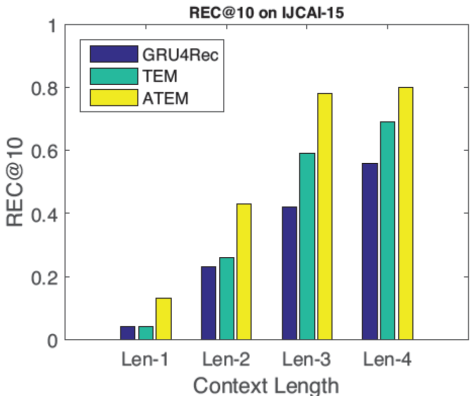
Discussions

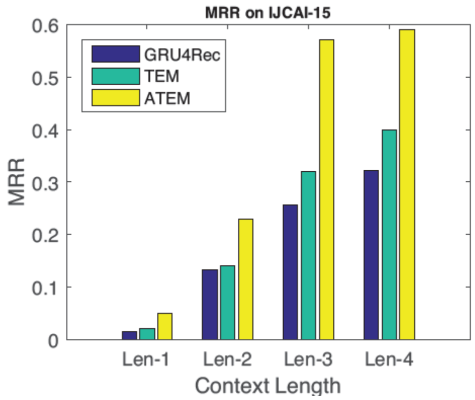
- ATEM clearly achieves the best results on both dataset.
- The highest MRR proves that ATEM can effectively put the users' desired items in the front of the recommendation list.
  - ATEM builds context from whole items while others only use first-order dependency.
  - Attention helps to find important item a lot.

Discussions

- ATEM has a very shallow and concise structure
  - Easy to train.
  - Efficient to recompute ranking scores while context keep updating.

- ATEM has more resistant for long context.





Discussions

- ATEM has more resistant for order shuffling.

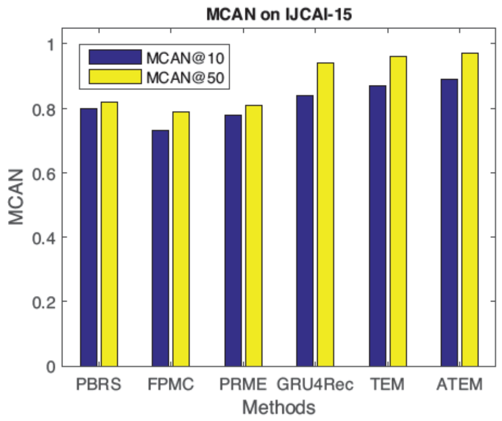
Table 4: Accuracy on disordered IJCAI-15

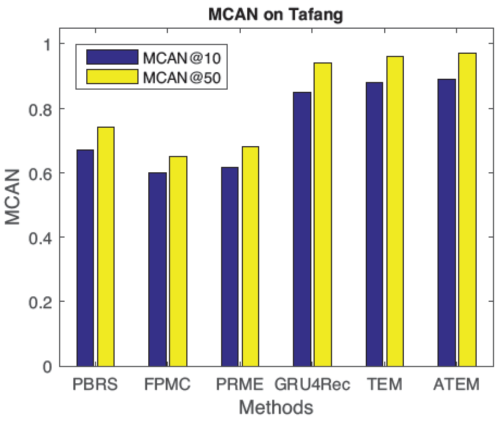
Model	REC@10	REC@50	MRR
PBRS	0.0500	0.0559	0.0185
FPMC	0.0151	0.0412	0.0183
PRME	0.0346	0.0389	0.0351
GRU4Rec	0.1636	0.2121	0.1022
ATEM	**0.3423**	**0.4981**	**0.1960**
TEM	0.2660	0.3012	0.1431

Discussions

- ATEM can generate novel recommendation list.
  - Novelty Metric MCAN@K

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





Conclusion

- ATEM is accurate, novel, efficient model for transactional context recommendation.
- Experiments show ATEM significantly beats other SotA models in real world datasets.
- We will explore the application of ATEM to other problems such as the author-topic relation learning.

Thank you !