

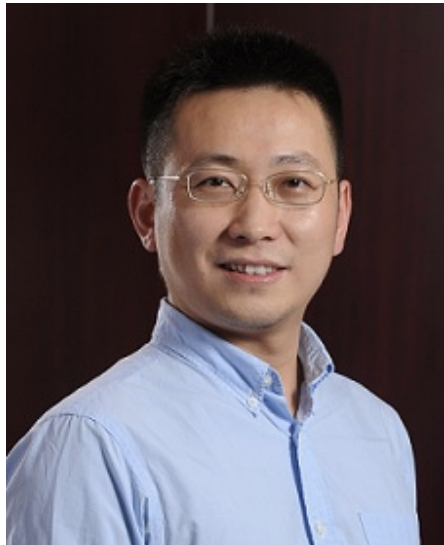
AI COLLEGE : RECOMMENDATION SYSTEM
INHWAN LEE

Session-based Recommendation with Graph Neural Networks

CONTENTS

1. Introduction
2. Model Overview
3. Session Graphs
4. Attention and Session Embeddings
5. Datasets and Results

AUTHORS



▶ Shu Wu



▶ Yuyuan Tang



▶ Yanqiao Zhu



▶ Liang Wang

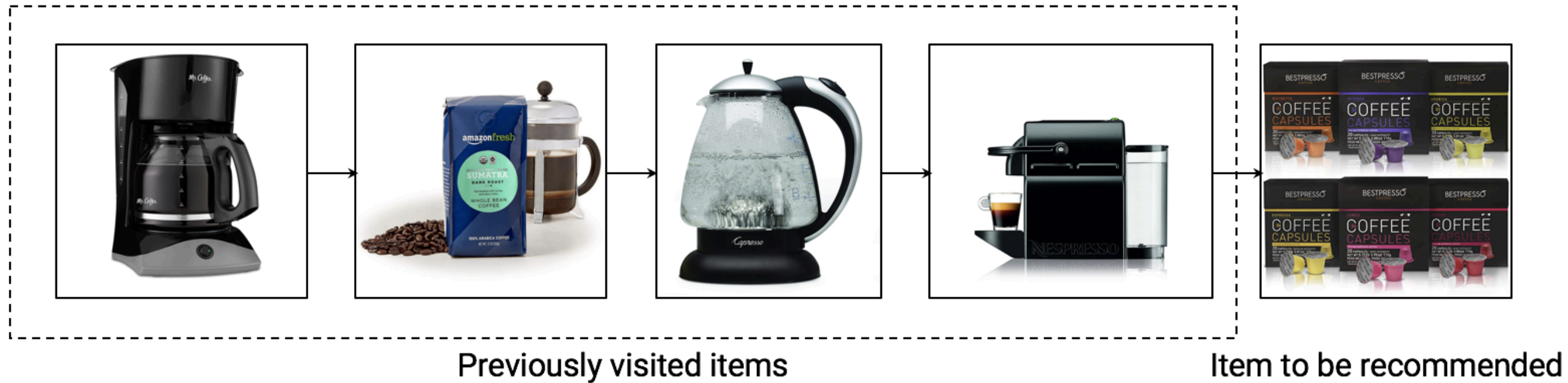


▶ Xing Xie



▶ Tieniu Tan

SESSION-BASED RECOMMENDATION



- ▶ 단기간에 유저가 선택한 아이템의 순서에 따라 다음 아이템을 추천
- ▶ 유저 고유의 정보는 반영하지 않음

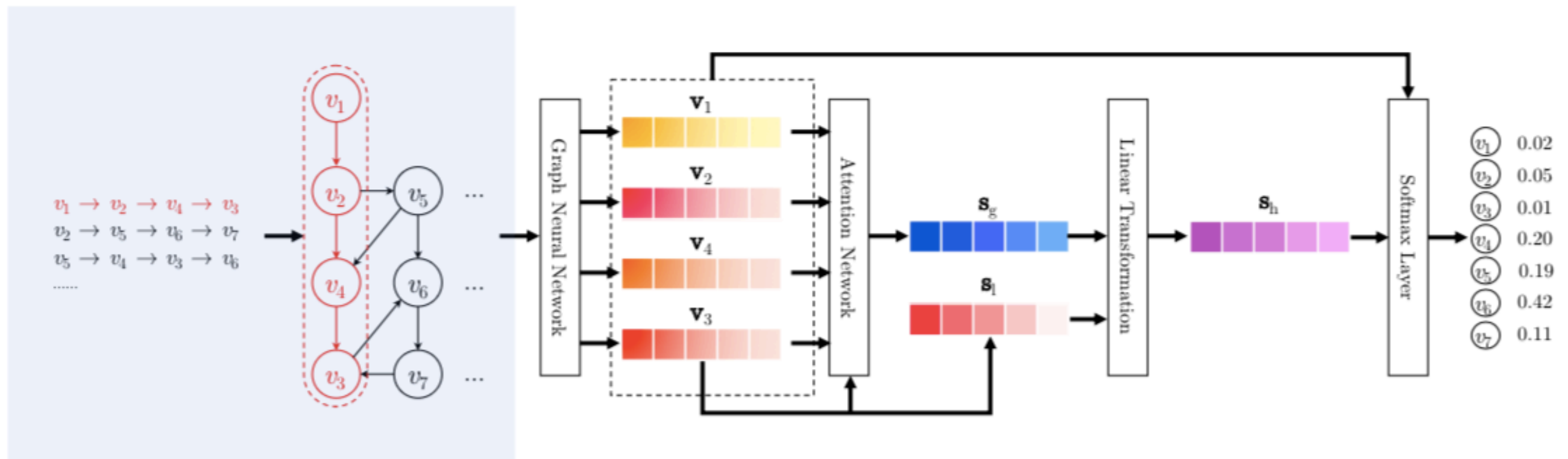
SESSION-BASED RECOMMENDATION

- ▶ Session-based Recommendation은 시계열 문제이므로, RNN (LSTM, GRU 등) 을 활용한 모델이 많이 등장
 - ▶ 그러나 이 모델들의 hidden vector는 유저 벡터로 간주되는데, 하나의 짧은 세션 만으로 유저의 특성을 담아내기 어렵음
 - ▶ 또한 RNN 모델은 연속적이지 않은 아이템 간의 관계를 포착하기 어려움
- ▶ 이를 해결하기 위해 그래프를 이용한 GNN 모델 또한 시도되어 왔음
- ▶ 이 논문에서는 그래프와 RNN, 그리고 Attention을 이용한 모델 소개

INPUT / OUTPUT OF MODEL

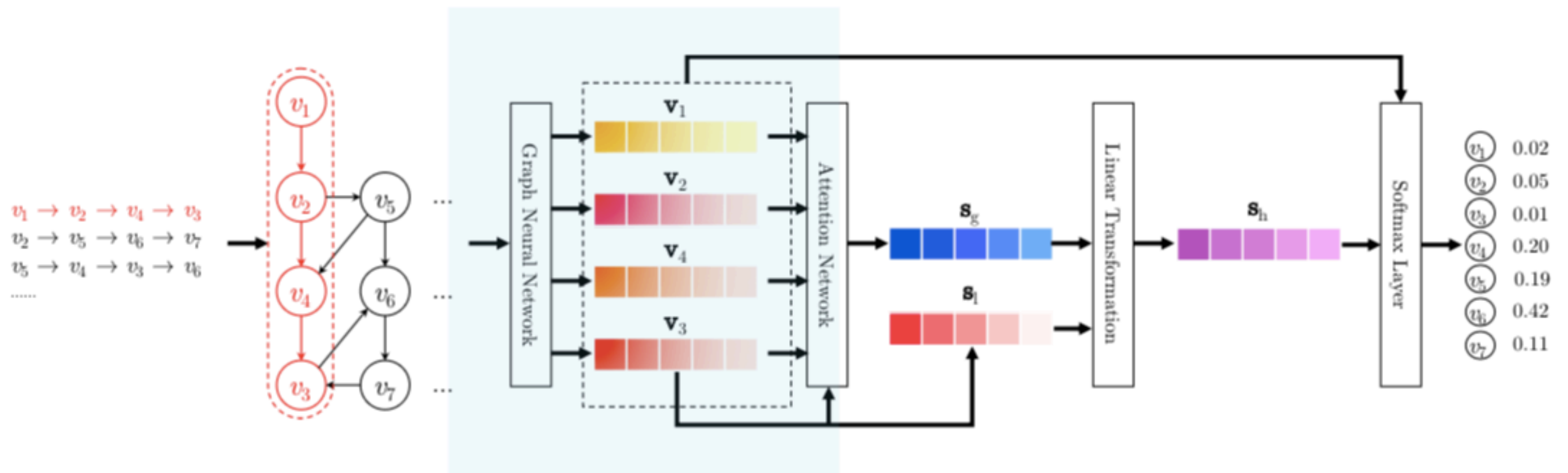
- ▶ 모든 아이템을 모은 집합을 $V = \{v_i\}_{i=1}^m$ 라고 하자.
그리고 각 아이템 v_i 의 벡터 표현 $\mathbf{v}_i \in \mathbb{R}^d$ 를 초기화하자.
- ▶ 한 유저가 하나의 세션 s 에서 클릭한 아이템을 순서대로 $s = [v_{s,1}, v_{s,2}, \dots, v_{s,n}]$ ($v_{s,i} \in V$) 라고 할 때,
SR의 목표는 다음 아이템 $v_{s,n+1}$ 을 예측하는 것이다.
- ▶ 논문의 모델은 세션 s 를 입력으로 받았을 때,
다음 아이템의 예측 확률 벡터 $(y_i)_{i=1}^m$ 를 출력한다.

SESSION GRAPH MODELING



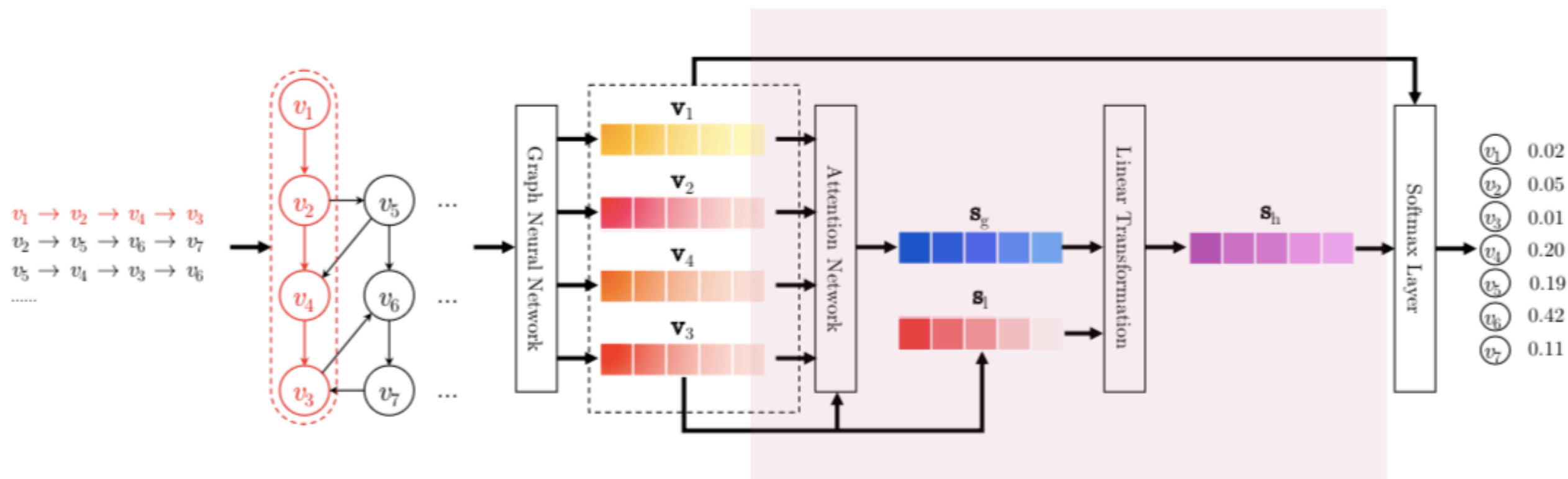
- ▶ 세션의 아이템을 이용하여 그래프를 생성한다

NODE REPRESENTATION LEARNING



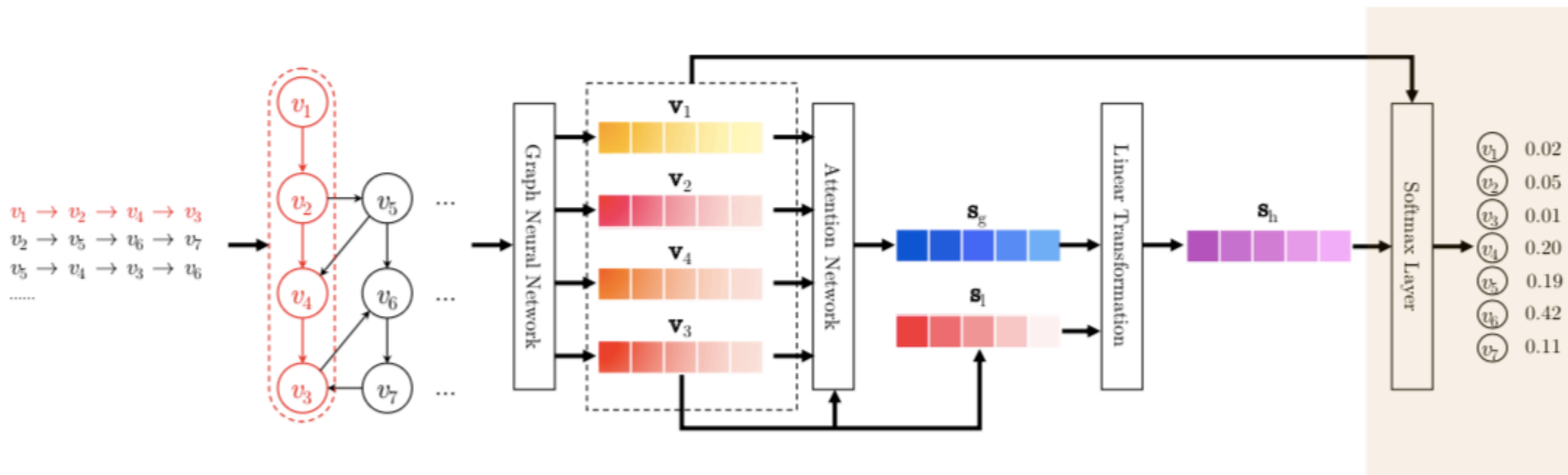
- ▶ 그래프를 이용하여 아이템들의 벡터 표현을 얻는다.

SESSION REPRESENTATION GENERATING



- ▶ Attention을 이용하여 세션의 벡터 표현을 얻는다.

MAKING RECOMMENDATION



- ▶ 아이템과 세션의 표현을 이용하여 최종 확률 벡터를 산출한다.

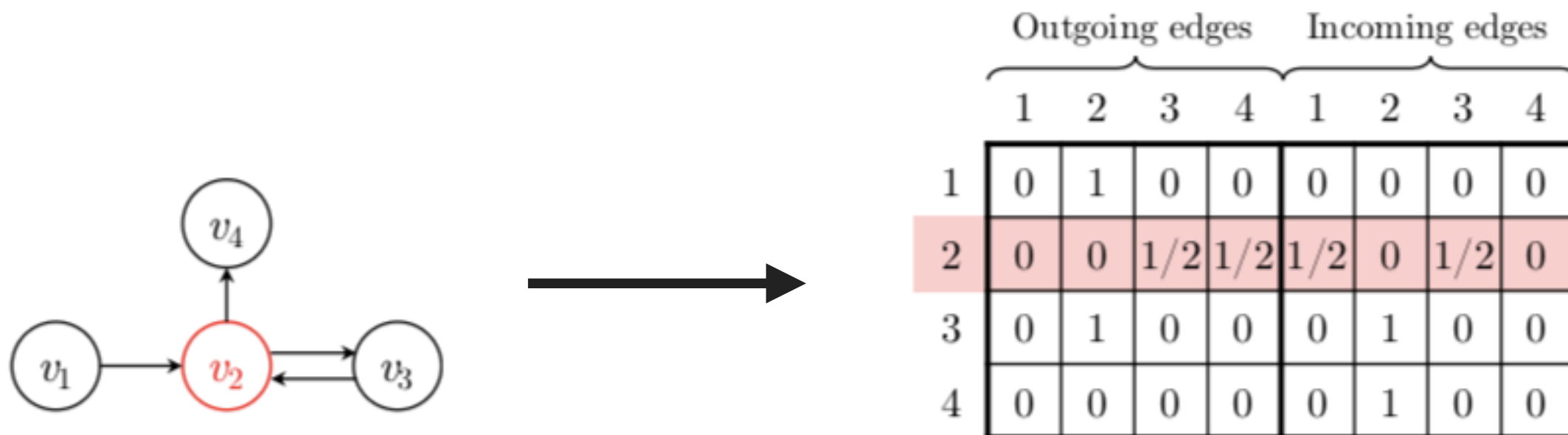
CONSTRUCTING SESSION GRAPHS

- ▶ 임의의 세션 s 는 다음과 같이 directed graph 로 표현할 수 있다.



CONNECTING MATRIX OF GRAPHS

- ▶ Directed graph는 각 점에서 나가는 선의 도착점과, 각 점으로 들어오는 선의 출발점을 이용하여 행렬로 표현할 수 있다.
- ▶ 또한 하나의 점에서 연결된 선이 여러개일 경우 전체 선의 갯수로 나눠 normalize 시켜준다.



- ▶ 이 행렬을 $A_s \in \mathbb{R}^{n \times 2n}$, 행렬의 i 열을 $A_{i,s} \in \mathbb{R}^{1 \times 2n}$ 로 표기한다.

LEARNING ITEM EMBEDDINGS ON GRAPHS

- ▶ 세션 $s = [v_1, \dots, v_n]$ 의 벡터 표현 $\mathbf{v}_i \in \mathbb{R}^d$ 을 어떻게 학습시킬 것인가?
 - ▶ 세션 그래프의 연결 행렬과 GRU-like한 순전파를 이용한다!

$$\mathbf{a}_{s,i}^t = \mathbf{A}_{s,i} [\mathbf{v}_1^{t-1}, \dots, \mathbf{v}_n^{t-1}]^\top \mathbf{H} + \mathbf{b},$$

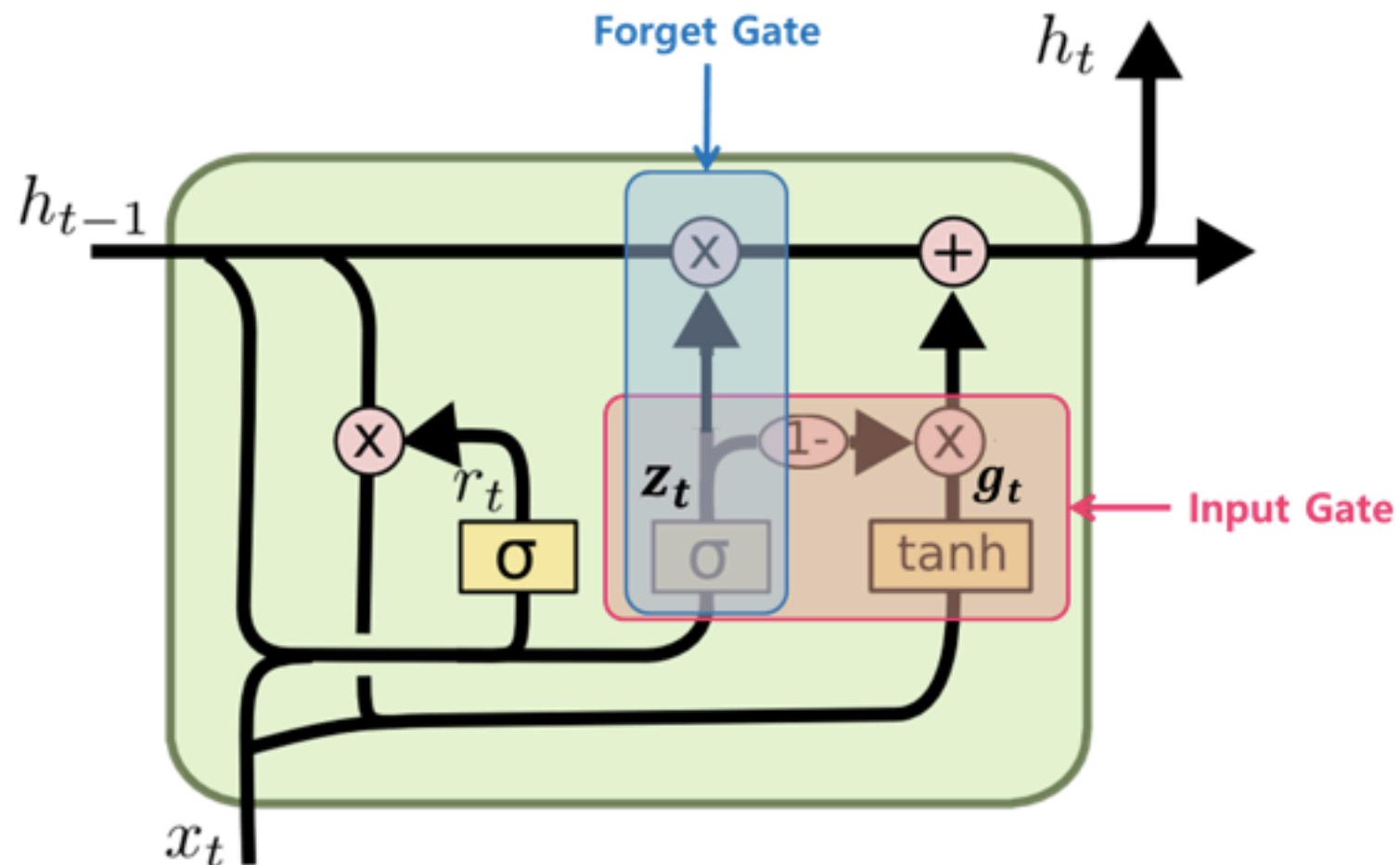
$$\mathbf{z}_{s,i}^t = \sigma(\mathbf{W}_z \mathbf{a}_{s,i}^t + \mathbf{U}_z \mathbf{v}_i^{t-1}), \quad \text{Reset gate}$$

$$\mathbf{r}_{s,i}^t = \sigma(\mathbf{W}_r \mathbf{a}_{s,i}^t + \mathbf{U}_r \mathbf{v}_i^{t-1}), \quad \text{Update gate}$$

$$\tilde{\mathbf{v}}_i^t = \tanh(\mathbf{W}_o \mathbf{a}_{s,i}^t + \mathbf{U}_o (\mathbf{r}_{s,i}^t \odot \mathbf{v}_i^{t-1})), \quad \text{Candidate}$$

$$\mathbf{v}_i^t = (1 - \mathbf{z}_{s,i}^t) \odot \mathbf{v}_i^{t-1} + \mathbf{z}_{s,i}^t \odot \tilde{\mathbf{v}}_i^t. \quad \text{Final representation}$$

CF. GRU STRUCTURE



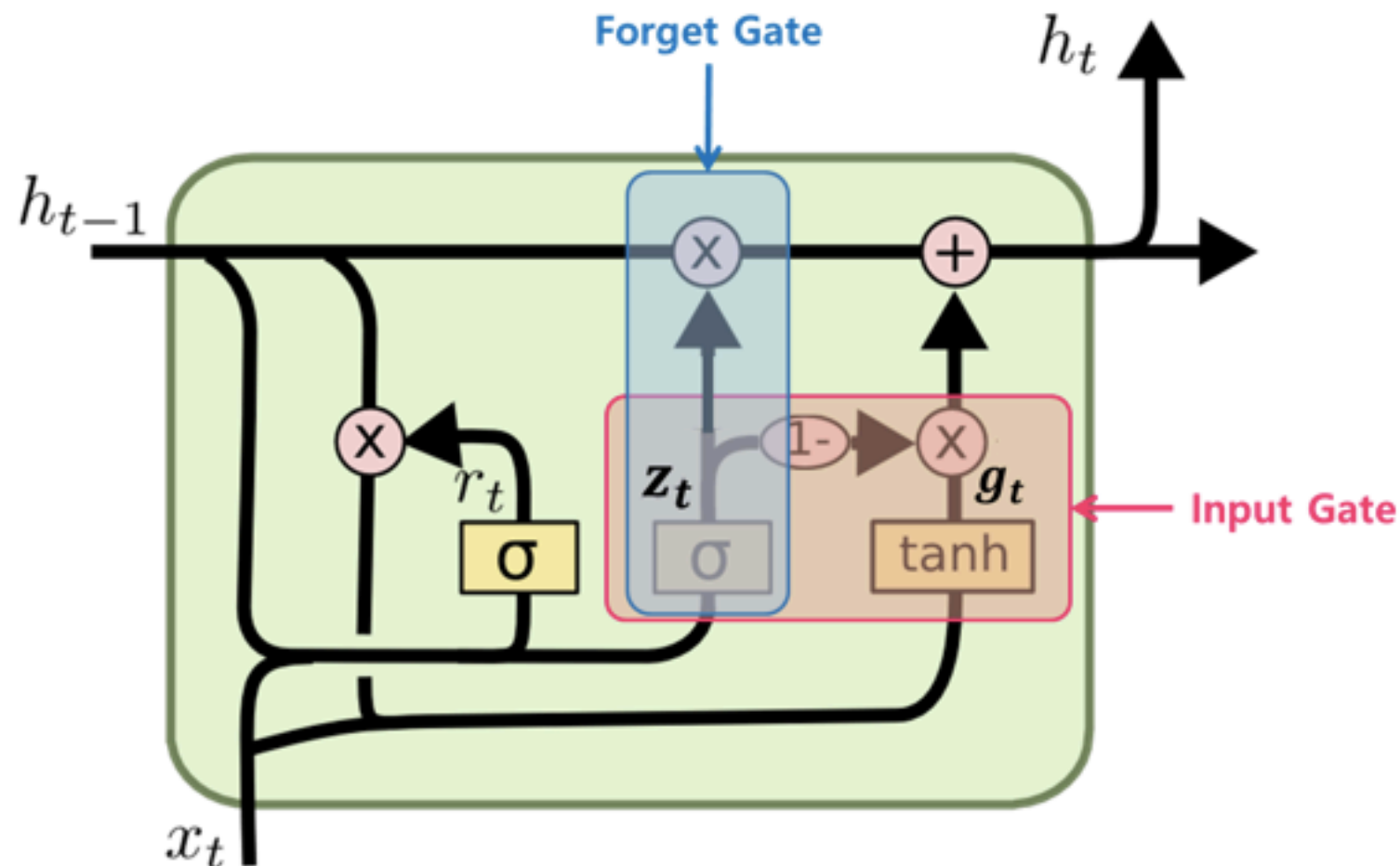
$$z_t = \sigma(x_t U_z + h_{t-1} W_z)$$

$$r_t = \sigma(x_t U_r + h_{t-1} W_r)$$

$$g_t = \tanh(x_t U_g + (h_{t-1} \odot r_t) W_g)$$

$$h_t = (1 - z_t) \odot g_t + z_t \odot h_{t-1}$$

CF. GRU STRUCTURE



$$z_t = \sigma(x_t U_z + h_{t-1} W_z)$$

$$r_t = \sigma(x_t U_r + h_{t-1} W_r)$$

$$g_t = \tanh(x_t U_g + (h_{t-1} \odot r_t) W_g)$$

$$h_t = (1 - z_t) \odot g_t + z_t \odot h_{t-1}$$

$$x_t = A_{s,i} [v_1^{t-1}, \dots, v_n^{t-1}]^T \mathbf{H} + \mathbf{b}, \quad h_t = \mathbf{v}_i^{t-1}$$

를 대입하면 논문의 수식을 얻을 수 있다.

LEARNING ITEM EMBEDDINGS ON GRAPHS

- ▶ 세션의 각 아이템의 벡터 표현을 이용하여 세션의 벡터 표현을 구한다.
- ▶ Local Embedding : 마지막으로 클릭한 아이템의 벡터 표현

$$\mathbf{s}_l = \mathbf{v}_n$$

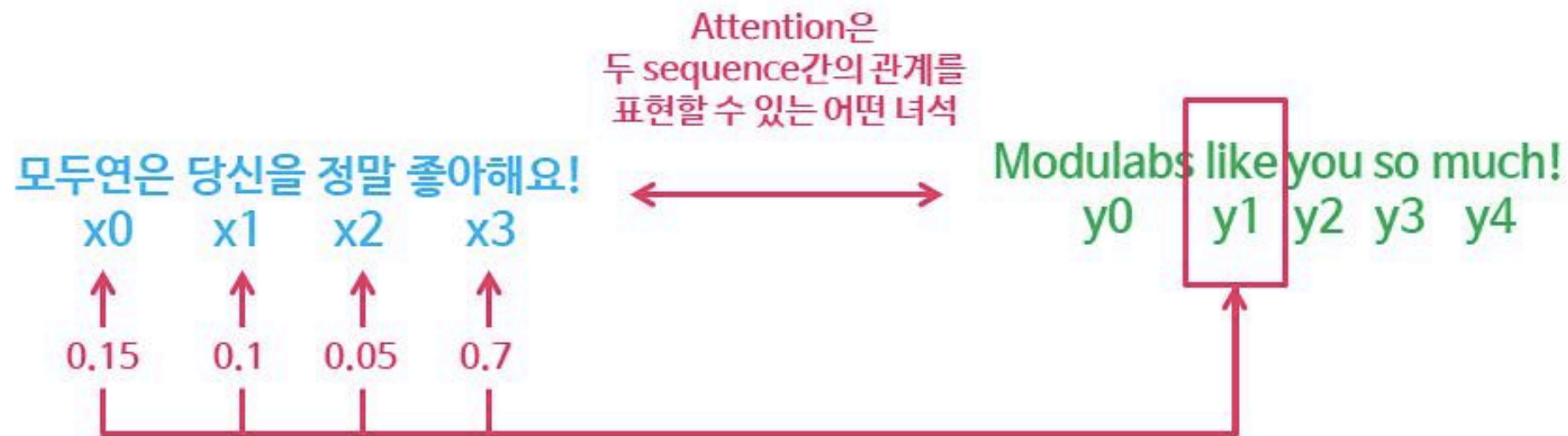
- ▶ Global Embedding : Attention 구조를 이용하여, 세션의 각 아이템이 세션의 벡터 표현에 각각 영향력을 발휘하도록 한다.

$$\alpha_i = \mathbf{q}^T \sigma(\mathbf{W}_1 \mathbf{v}_n + \mathbf{W}_2 \mathbf{v}_i + \mathbf{c}) : \text{attention value}$$

$$\mathbf{s}_g = \sum_{i=1}^n \alpha_i \mathbf{v}_i$$

ATTENTION MECHANISM

- ▶ Attention mechanism의 핵심 아이디어
 - ▶ 순차적인 데이터를 입력으로 받아서 결과값을 예측할 때, 해당 시점에서 예측해야 할 값과 연관이 있는 입력 부분을 좀 더 집중(attention)해서 본다.



- ▶ 세션의 global embedding을 학습할 때, 실제 다음 아이템의 예측에 가장 큰 영향을 준 아이템에 가중치를 주어 학습할 수 있다!

HYBRID EMBEDDING AND RECOMMENDATION

- ▶ 세션의 local/global embedding을 연결하여 hybrid embedding 산출

$$\mathbf{s}_h = \mathbf{W}_3[\mathbf{s}_l; \mathbf{s}_g]$$

- ▶ 세션 s 의 다음 아이템 예측을 위한 확률 벡터 $\hat{\mathbf{y}}$ 는 세션 벡터와 전체 아이템의 벡터 표현을 내적하여 구한다.

$$\hat{\mathbf{z}} = (\hat{\mathbf{z}}_i) = (\mathbf{s}_h^T \cdot \mathbf{v}_i)_{i=1}^m$$

$$\hat{\mathbf{y}} = \textit{softmax}(\hat{\mathbf{z}})$$

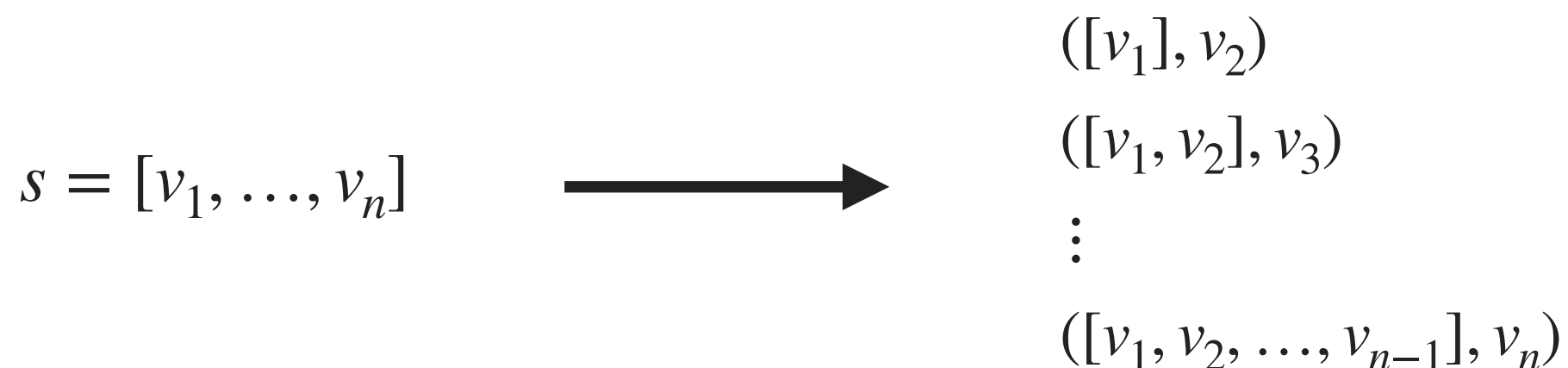
- ▶ 모델의 손실 함수는 크로스 엔트로피를 이용한다.

DATASETS

- ▶ Datasets
 - ▶ Yoochoose from RecSys Challenge 2014
 - ▶ Diginetica from CIKM Cup 2016
- ▶ Data Structure

session_id	user_id	item_id	timeframe	eventdate
------------	---------	---------	-----------	-----------

- ▶ Data Augmentation



RESULTS

Method	Yoochoose 1/64		Yoochoose 1/4		Diginetica	
	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
POP	6.71	1.65	1.33	0.30	0.89	0.20
S-POP	30.44	18.35	27.08	17.75	21.06	13.68
Item-KNN	51.60	21.81	52.31	21.70	35.75	11.57
BPR-MF	31.31	12.08	3.40	1.57	5.24	1.98
FPMC	45.62	15.01	—	—	26.53	6.95
GRU4REC	60.64	22.89	59.53	22.60	29.45	8.33
NARM	68.32	28.63	69.73	29.23	49.70	16.17
STAMP	68.74	29.67	70.44	30.00	45.64	14.32
SR-GNN	70.57	30.94	71.36	31.89	50.73	17.59

EXPERIMENT

```
-----  
epoch: 13  
start training: 2020-01-30 02:51:51.908625  
[0/7195] Loss: 4.1552  
[1440/7195] Loss: 4.1679  
[2880/7195] Loss: 4.1507  
[4320/7195] Loss: 3.6669  
[5760/7195] Loss: 3.7158  
Loss:28188.936  
start predicting: 2020-01-30 02:56:55.770174  
Best Result:  
    Recall@20: 51.4542    MMR@20: 17.8041  
-----  
Run time: 4484.156018 s
```

RELATION BETWEEN NLP AND RS

- ▶ Attention은 자연어처리에서 처음 도입 (GRU + Attention)
(Neural Machine Translation by Jointly Learning to Align and Translate, 2014)
- ▶ RNN 이 가지는 한계를 극복하기 위해 attention만으로
순차적 데이터를 모델링한 Transformer의 도입
(Attention Is All You Need, 2017)
- ▶ 양방향 Transformer 구조를 통해 더 깊은 구조를 학습한 BERT 모델
(BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2019)

RELATION BETWEEN NLP AND RS

- ▶ Session-based Recommendation + GRU + Attention = SR-GNN
- ▶ Session-based Recommendation + Transformer
(Behavior Sequence Transformer for E-commerce Recommendation in Alibaba, 2019)
- ▶ Session-based Recommendation + BERT
(BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer, 2019)

RELATION BETWEEN NLP AND RS

- ▶ Session-based Recommendation + GRU + Attention = SR-GNN
- ▶ Session-based Recommendation + Transformer
(Behavior Sequence Transformer for E-commerce Recommendation in Alibaba, 2019)
- ▶ Session-based Recommendation + BERT
(BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer, 2019)

and more...?

감사합니다!