Recommendation Tech seminar

염은지

Contents

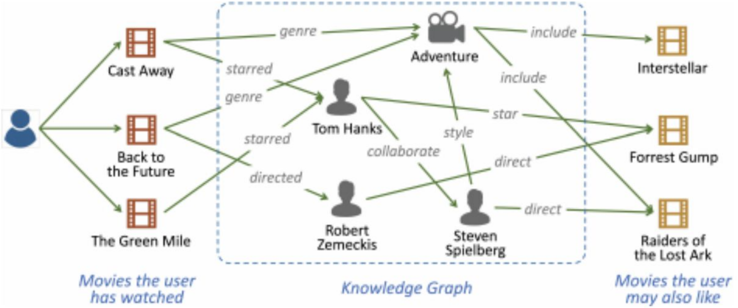
추천 시스템(RS)이란

- 온라인 환경에서 사용자의 선호(피드백)을 이용하여 취향을 고려한 아이템을 추천해주는 시스템
- 높은 정확도에 기반한 추천으로 사용자의 의사결정 과정을 돕는 것이 목적

E-business	추천 대상
YouTube	시청할 가능성이 높은 Video
Facebook	관계를 맺을 확률이 높은 사람
Amazon, eBay 등의 E-commerce 산업	좋아하거나 구입할 수 있는 상품
News 산업	읽을 확률이 높은 뉴스
Glassdoor 등의 job matching 산업	어울리는 직업
TripAdvisor 등의 여행 산업	적합한 휴가지

추천 접근 방식

- **Collaborative filtering recommender system (CFRS)**
  - user의 feedback history를 추적하여 유사한 history를 가진 다른 user에 기반하여 item 추천
- **Content-based recommender system (CBRS)**
  - user profile과 item description(attribute)을 matching
- **Demographic recommendation system (DRS)**
  - users' demographic profile에 기반한 추천
- **Hybrid recommender system (HRS)**
  - 여러 filtering 방식 결합
- **Knowledge-based recommender system (KBRS)**
  - user의 preference와 item들의 attribute를 고려하여 추



- **Context-aware recommender system (CARS)**
  - user-item interaction 이외의 context까지 고려 $R: user \times item \times context \rightarrow rating$

추천 시스템의 평가

- **offline basic metric**
  - **no rank-aware metrics**
    - accuracy
    - precision, recall, f1-score
    - precision@K, recall@K
    - Top-N Hit rate
  - **rank-aware metrics**
    - MRR(mean reciprocal rank)
    - MAP(mean average precision)
    - NDCG(normalized discounted cumulative gain)

- **offline additional metric**
  - coverage
  - popularity
  - serendipity
  - diversity
  - personalization

- **online metric**
  - A/B test
  - lab experiment
  - survey

ACM RecSys 2021, KDD 2021로 본 추천 기술 현황

- **Tech**
  - reinforcement learning 기술 활용
  - Contextual Bandit 논문
  - CF, CB 기반 방법론에 다양한 DL 기술/방법 접목 및 개선
    - autoencoder와 graph structure
    - transfer learning, siamese networks 등...
- **Topic**
  - sequential data에 대한 context-aware 및 embedding 표현
  - bias reduction
  - evaluation
  - large scale
  - explainable AI

- **Domain**
  a. music
  b. e-commerce
  c. advertisement
  d. education
  e. ....

ACM RecSys 2021, KDD 2021로 본 추천 논문 현황

- 요약
  - Online recommendation, Conversational Recommendation을 위한 기술 연구
  - Context를 추출하여 추천에 적용하기 위한 방법 연구
  - 추천에 쓰였던 기존의 방법론들과 Deep learning 기술 hybriding을 통한 성능 향상
  - 추천 서비스 개선/고도화를 위한 다양한 주제 실험

논문 선정 이유

- **sequential data modeling 및 user context 추출**
  - **Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation**
    - 문장/단어의 embedding을 추천에서 sequential한 interaction 표현으로 치환
    - NLP에서 사용되던 모델링 기법을 context 추출 및 inference에도 적용 가능
  - **Global-Local Item Embedding for Temporal Set Prediction**
    - 특정 시점의 예측에서 개인(local) 이외의 다른 유저들(global)한 정보까지 함께 고려
    - 시간 순서에 따른 item embedding을 이용하여 user의 temporal한 context를 추출 가능

논문 선정 이유

- **sequential data modeling 및 user context 추출**
  - **Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation**
    - 문장/단어의 embedding을 추천에서 sequential한 interaction 표현으로 치환
    - NLP에서 사용되던 모델링 기법을 context 추출 및 inference에도 적용 가능
  - **Global-Local Item Embedding for Temporal Set Prediction**
    - 특정 시점의 예측에서 개인(local) 이외의 다른 유저들(global)한 정보까지 함께 고려
    - 시간 순서에 따른 item embedding을 이용하여 user의 temporal한 context를 추출 가능

Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation

- Transformer 계열 모델을 Sequence/Session based recommendation에 적용
- HuggingFace의 Transformers library 을 기반으로 한 open source
- https://github.com/NVIDIA-Merlin/Transformers4Rec/

Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation

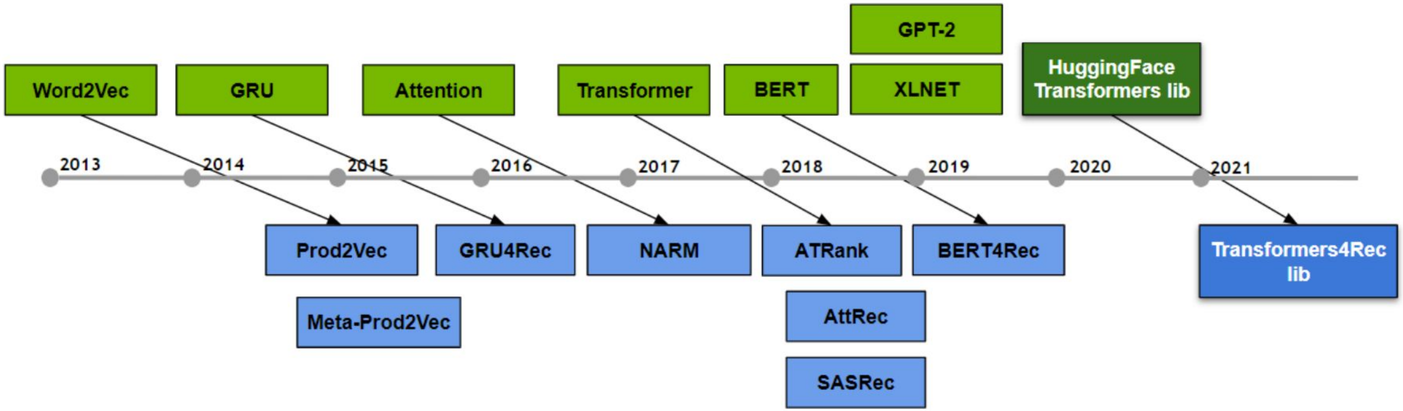


Figure 1: A timeline illustrating the influence of NLP research in Recommender Systems

Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation

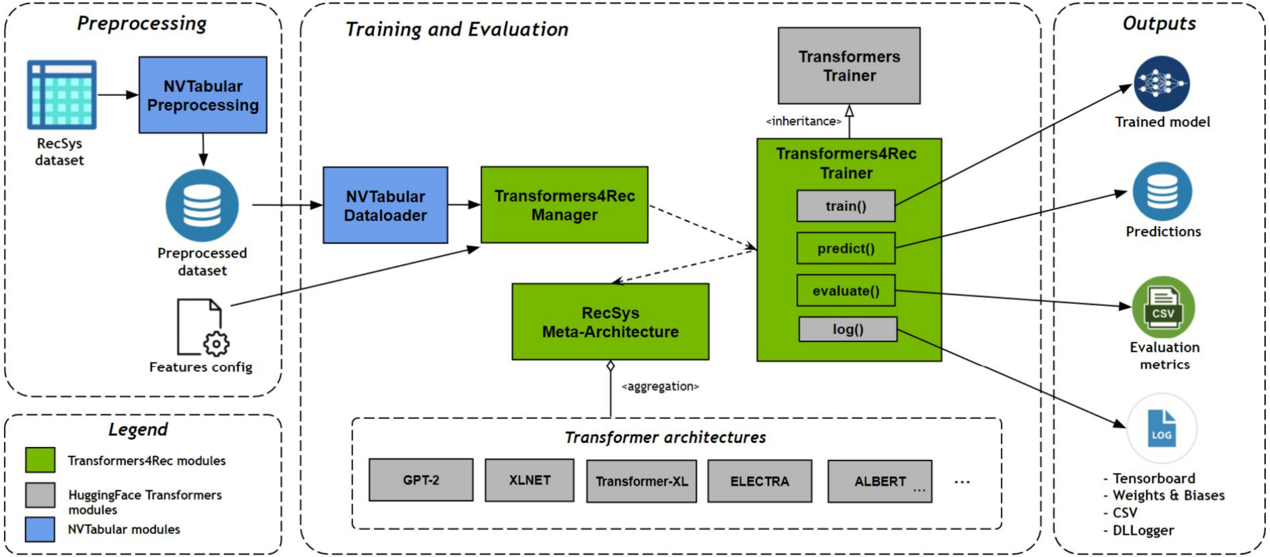


Figure 2: Transformers4Rec pipeline overview

Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation

- Python, PyTorch 사용
- Data preprocessing and feature engineering
  - NVIDIA NVTabular library 사용
- Model training and evaluation
  - Meta-Architecture는 highly configurable component
  - Top-N ranking metrics 사용
  - incremental training 가능

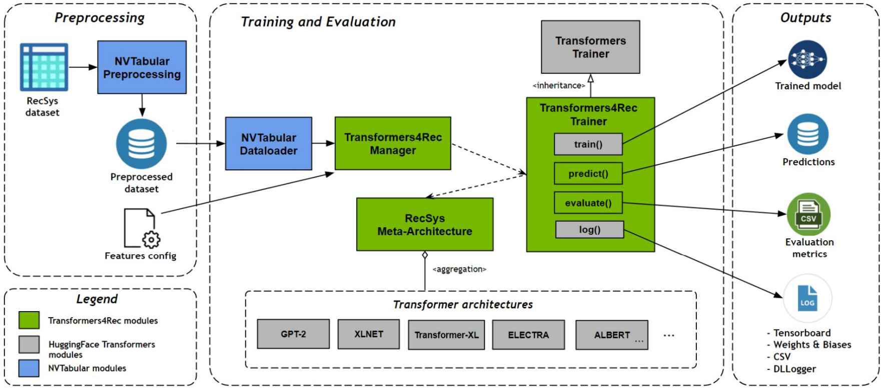


Figure 2: Transformers4Rec pipeline overview

Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation

- **Meta-Architecture**
    a. Features Processing module
    b. Sequence Masking module
    c. Sequence Processing module
    d. Prediction head module

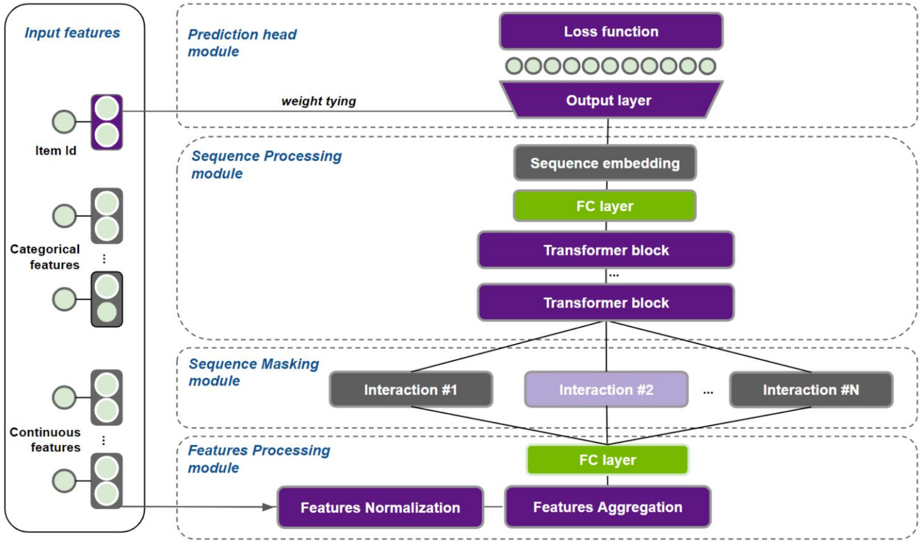


Figure 3: Transformers4Rec neural meta-architecture

Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation

- **Input Feature Representation**
- **Input Feature Aggregation**
  - concatenation merge
  - element-wise merge

$$m_k = \text{concat}(x_k^{(u)}, f_{1,k}^{(u)}, \ldots, f_{I,k}^{(u)})$$

$$m_k = x_k^{(u)} \odot [1 + f_{1,k}^{(u)} + \ldots + f_{I,k}^{(u)}]$$

- **Tying Embeddings**
  - input embedding weight와 output projection layer matrix를 같은 공간으로 사용하여 model parameter 감소
- **Regularization**
  - Label Smoothing이 train, validation accuracy 개선에 가장 좋은 효과
- **Loss functions**
  - cross-entropy 사용 (pairwise loss도 사용 가능)
- **Extensibility**
  - 여러 개 sequence input을 별도의 Transformer에서 training 후 output 결합 가능
  - custom prediction head 생성 가능

Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation

- methodology
  - Incremental Training and Evaluation
    - day/hour 단위
    - validation : test = 50 : 50
    - $T_1, \ldots, T_i \rightarrow T_{i+1}$
  - Hyperparameter optimization
    - maximizing NDCG@20
    - 특정 날짜 overfit 방지 위해 유효한 날짜의 50%만 training, evaluation
  - Metrics
    - best hyperparameters로 5번 실행하여 평균 낸 metric 사용
    - primary metric : NDCG@20
    - sub metric : HR@20
  - Datasets and Preprocessing
  - Baseline Algorithms
    - KNN 계열 알고리즘, GRU4Rec

Table 1: Dataset statistics

Dataset	days	items (K)	sessions (M)	interactions (M)	sessions length (avg.)	Gini index
REES46 eCommerce	31	156,516	3,268,268	17,967,918	5.49	0.86
YOOCHOOSE eCommerce	182	50,549	6,756,575	26,478,390	3.83	0.89
G1 news	16	46,027	1,048,556	2,988,037	2.84	0.94
ADRESSA news	16	13,820	982,210	2,648,999	2.69	0.96

Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation

Table 2: Experimental Results: RQ1 / RQ2 / RQ3

		REES46 eCommerce		YOOCHOOSE eCommerce		G1 news		ADRESSA news	
	Algorithm	**NDCG@20**	**HR@20**	**NDCG@20**	**HR@20**	**NDCG@20**	**HR@20**	**NDCG@20**	**HR@20**
RQ1	V-SkNN	0.2187	0.4662	0.2975	0.5110	0.3511	0.6601	0.3590	0.7210
	STAN	0.2194	0.4797	0.3082	0.5196	0.3570	**0.6681**	0.3635	0.7246
	VSTAN	0.2200	**0.4857***	0.3097	0.5206	0.3586	0.6668	0.3617	0.7241
	GRU4Rec (FT)	0.2231	0.4414	0.3442	0.5891	0.2596	0.5029	0.3007	0.6052
	GRU4Rec (SWT)	0.2204	0.4359	0.3431	0.5885	0.2666	0.5183	0.2967	0.5948
	GRU (CLM)	0.2139	0.4315	0.2975	0.6129	0.3549	0.6632	0.3799	**0.7413**
	GPT-2 (CLM)	0.2165	0.4338	0.2975	0.6065	0.3560	0.6620	0.3790	0.7398
	Transformer-XL (CLM)	0.2197	0.4404	0.3585	0.6133	0.3294	0.6192	**0.3811***	0.7382
	BERT (MLM)	0.2218	0.4672	**0.3750***	**0.6349***	0.3549	0.6549	0.3725	0.7221
	ELECTRA (RTD)	**0.2430**	0.4768	0.3722	0.6294	**0.3588**	0.6600	0.3729	0.7226
	XLNet (PLM)	0.2422	0.4760	0.3681	0.6282	0.3551	0.6634	0.3673	0.7212
RQ2	XLNet (PLM) - original	0.2422	0.4760	0.3681	0.6282	0.3551	**0.6634***	0.3673	0.7212
	XLNet (CLM)	0.2108	0.4219	0.3557	0.6079	0.3551	0.6508	0.3770	**0.7378***
	XLNet (RTD)	**0.2546***	**0.4886***	**0.3776**	0.6373	**0.3609**	0.6611	0.3816	0.7329
	XLNet (MLM)	0.2428	0.4763	**0.3776**	**0.6384***	0.3607	0.6605	**0.3822**	0.7349
RQ3	XLNet (MLM) - item id	0.2428	0.4763	0.3776	0.6384	0.3607	0.6605	0.3822	0.7349
	Concat. merge	0.2522	0.4782	-	-	0.3652	0.6714	**0.3912***	**0.7488***
	Concat. merge + SOHE	**0.2542***	**0.4858**	-	-	**0.3675***	**0.6721***	0.3886	0.7463
	Element-wise merge	0.2529	0.4854	-	-	0.3614	0.6678	0.3892	0.7433

Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation

- **Q1**
  - Transformer 기반 모델이 모든 데이터세트에 잘 동작하지는 않음
  - 긴 session len을 갖고있는 commerce dataset의 NDCG의 향상이 더 높음 (+8.95%)

		REES46 eCommerce		YOOCHOOSE eCommerce		G1 news		ADRESSA news	
	Algorithm	**NDCG@20**	**HR@20**	**NDCG@20**	**HR@20**	**NDCG@20**	**HR@20**	**NDCG@20**	**HR@20**
RQ1	V-SkNN	0.2187	0.4662	0.2975	0.5110	0.3511	0.6601	0.3590	0.7210
	STAN	0.2194	0.4797	0.3082	0.5196	0.3570	**0.6681**	0.3635	0.7246
	VSTAN	0.2200	**0.4857***	0.3097	0.5206	0.3586	0.6668	0.3617	0.7241
	GRU4Rec (FT)	0.2231	0.4414	0.3442	0.5891	0.2596	0.5029	0.3007	0.6052
	GRU4Rec (SWT)	0.2204	0.4359	0.3431	0.5885	0.2666	0.5183	0.2967	0.5948
	GRU (CLM)	0.2139	0.4315	0.2975	0.6129	0.3549	0.6632	0.3799	**0.7413**
	GPT-2 (CLM)	0.2165	0.4338	0.2975	0.6065	0.3560	0.6620	0.3790	0.7398
	Transformer-XL (CLM)	0.2197	0.4404	0.3585	0.6133	0.3294	0.6192	**0.3811***	0.7382
	BERT (MLM)	0.2218	0.4672	**0.3750***	**0.6349***	0.3549	0.6549	0.3725	0.7221
	ELECTRA (RTD)	**0.2430**	0.4768	0.3722	0.6294	**0.3588**	0.6600	0.3729	0.7226
	XLNet (PLM)	0.2422	0.4760	0.3681	0.6282	0.3551	0.6634	0.3673	0.7212

Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation

- **Q2**
  - XLNet (RTD) 방식이 모든 dataset에서 가장 높은 NDCG@20을 가지며 HR@20도 높은 성능을 보임
  - MLM 방식이 모든 데이터세트에 대해서 CLM보다 우수
- **Q3**
  - additional feature를 추가하는 모든 경우에 대해서 성능 향상
  - SOHE(Soft-One Hot Encoding) 방법을 함께 쓰면 더욱 효과적

RQ2	XLNet (PLM) - original	0.2422	0.4760	0.3681	0.6282	0.3551	**0.6634***	0.3673	0.7212
	XLNet (CLM)	0.2108	0.4219	0.3557	0.6079	0.3551	0.6508	0.3770	**0.7378***
	XLNet (RTD)	**0.2546***	**0.4886***	**0.3776**	0.6373	**0.3609**	0.6611	0.3816	0.7329
	XLNet (MLM)	0.2428	0.4763	**0.3776**	**0.6384***	0.3607	0.6605	**0.3822**	0.7349
RQ3	XLNet (MLM) - item id	0.2428	0.4763	0.3776	0.6384	0.3607	0.6605	0.3822	0.7349
	Concat. merge	0.2522	0.4782	-	-	0.3652	0.6714	**0.3912***	**0.7488***
	Concat. merge + SOHE	**0.2542***	**0.4858**	-	-	**0.3675***	**0.6721***	0.3886	0.7463
	Element-wise merge	0.2529	0.4854	-	-	0.3614	0.6678	0.3892	0.7433

Global-Local Item Embedding for Temporal Set Prediction

- **Global-Local Item Embedding (GLOIE) 제안**
    - global/local information을 병합한 temporal properties을 이용하여 특정 시점에 대한 set prediction 진행
        - global embedding : Variational Autoencoder + tweedie distribution
        - local embedding : Dynamic graph-based model (DNNTSP)
        - attention method를 사용해 embedding 통합

Global-Local Item Embedding for Temporal Set Prediction

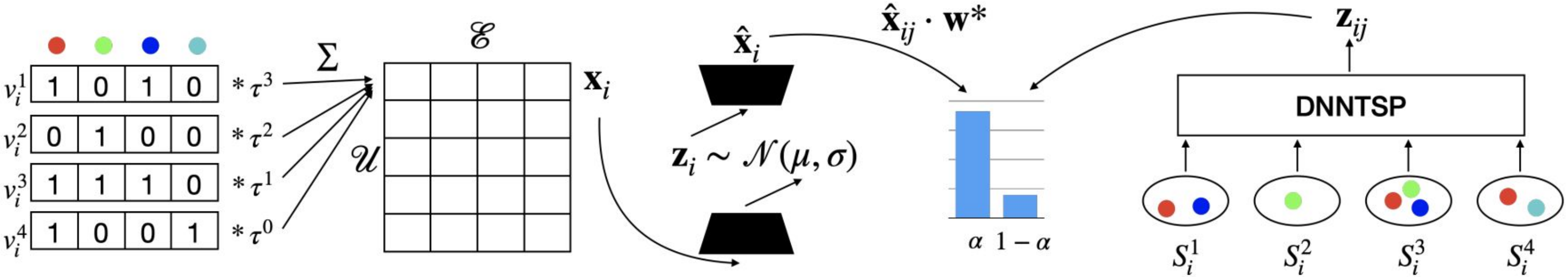
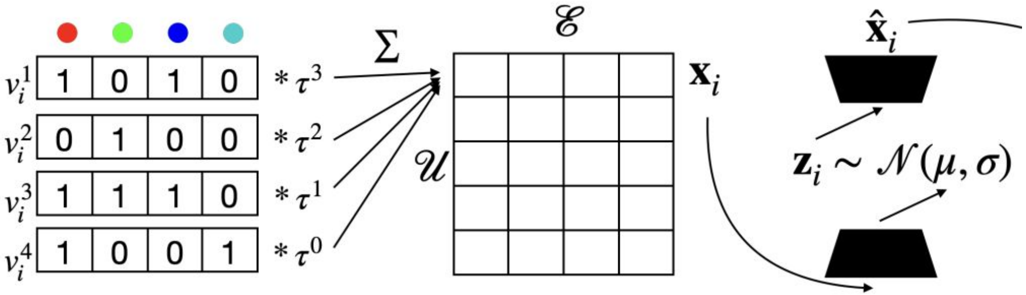


Fig. 1. Overview of GLOIE. We first make sum of time decayed vector $\mathbf{x}_i$ (left part of the figure). VAE maximizes the ELBO of each $\mathbf{x}_i$. The weighted sum of $\hat{\mathbf{x}}_{ij} \cdot \mathbf{w}^*$ and $\mathbf{z}_{ij}$ becomes final embedding for interacted items. The weight α is determined by the attention mechanism. For the items that the user never interacted, we just use $\hat{\mathbf{x}}_{ij}$.

Global-Local Item Embedding for Temporal Set Prediction

- **Learning Global-Local Information by VAE**
  - given user's history 뿐만 아니라 other user's history도 modeling
  - user마다 set의 sequence length가 다름 -> time decayed sequence of sets 적용



$$\mathbf{x}_i = \sum_{1 \le k \le T_i} \mathbf{v}_i^k \cdot \tau^{T_i - k}$$

$$\mathbf{z}_i \sim q(z \mid \mathbf{x}_i), \quad \hat{\mathbf{x}}_i \sim p(\mathbf{x}_i \mid \mathbf{z}_i)$$

$$\mathcal{L}_{vae} = \mathbb{E}_{\mathbf{x}_i \sim \mathcal{D}_u} \left[-\mathbb{E}_{\mathbf{z}_i \sim q(\mathbf{z}|\mathbf{x}_i)} [\log p(\mathbf{x}_i \mid \mathbf{z}_i)] + D_{KL}(q(\mathbf{z} \mid \mathbf{x}_i) || p(\mathbf{z})) \right]$$

Global-Local Item Embedding for Temporal Set Prediction

- **Tweedie Output on Decoder**
  - decay factor가 적용된 dataset의 분포는 zero-inflated / long-tailed
  - Decoder 출력 분포를 Tweedie distribution으로 설정
  - Tweedie distribution의 mean parameter μ와 power parameter p를 학습하여 MLE 수행
    - Tweedie distribution
      - a special case of exponential dispersion model (EDM)
      - 다음과 같은 property를 가지고 있 $V(\mu) = \mu^p$
      - $1 < p < 2$인 경우 compound poisson - gamma distribution 를 가짐

Consider two step sampling process $N \sim Poisson(\lambda)$ and $X_i \sim Gamma(\alpha, \beta)$ for $\lambda, \alpha, \beta > 0$ and $i = 1, \ldots, N$.
a random variable Z as follows:

$$Z = \begin{cases} 0 & N = 0 \\ X_1 + \cdots + X_N & N > 0 \end{cases}$$

$$\mathcal{L}_{Tweedie}(z, \mu, p) = -z \cdot \frac{\mu^{1-p}}{1-p} + \frac{\mu^{2-p}}{2-p}$$

Global-Local Item Embedding for Temporal Set Prediction

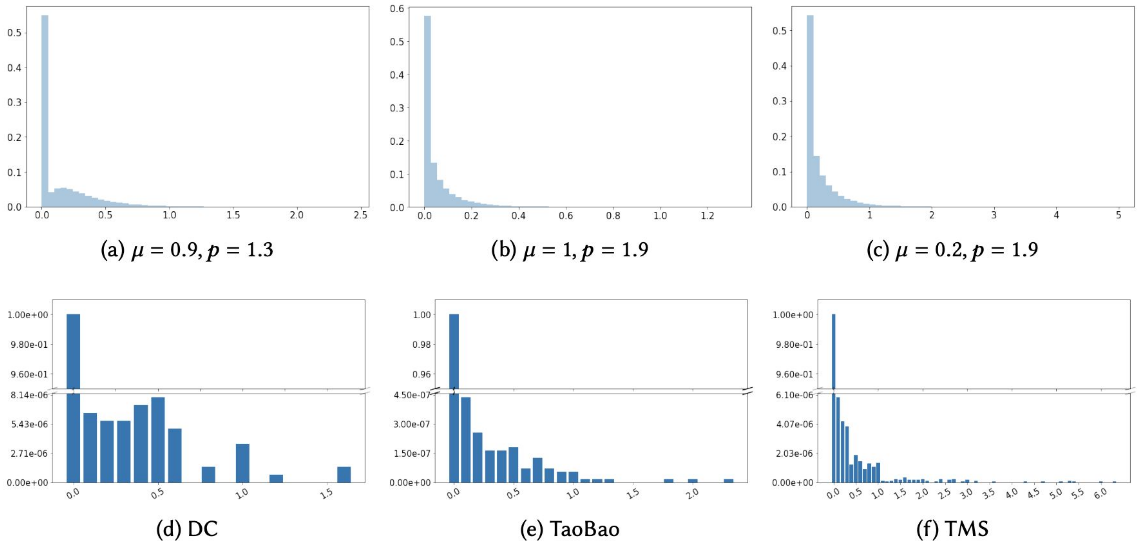
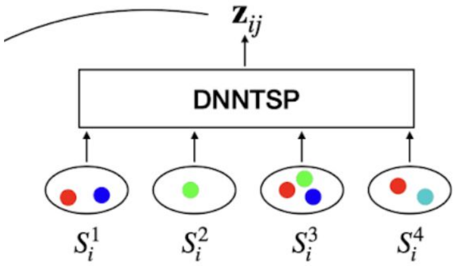
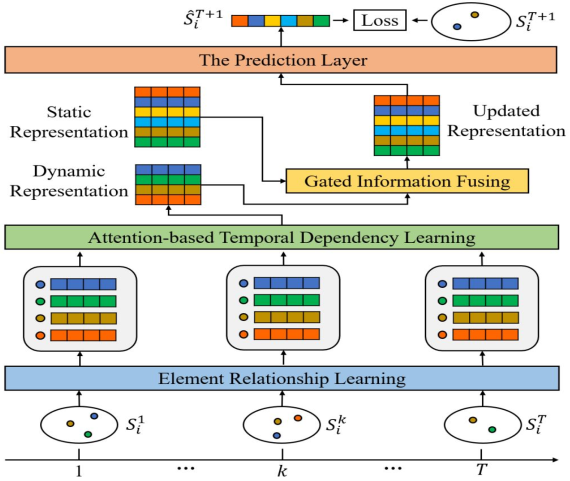


Fig. 2. Upper row: Histogram plot of samples from Tweedie distribution. Lower row: Histogram plot of elements of sum of decayed vector $\mathbf{x}_i$. Across all benchmarks, the distributions are zero-inflated and long-tailed.

Global-Local Item Embedding for Temporal Set Prediction

- **Integrating Global-Local Information**
  - Tweedie output을 갖는 VAE는 빈번한 interaction item에 대한 user preference를 과소 추정하는 경향이 있음
  - DNNTSP을 활용하여 자주 interaction하는 item embec
    - Element Relationship Learning(ERL)
    - Temporal Dependency Learning (TDL)
    - Gated Information Fusing

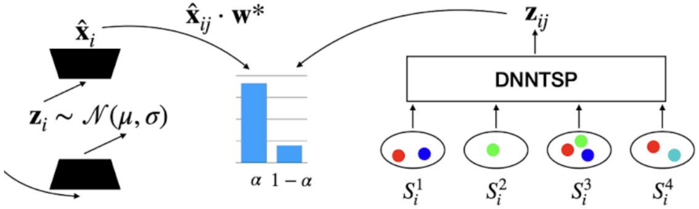




Global-Local Item Embedding for Temporal Set Prediction

- **Integrating Global-Local Information**
  - $\hat{x}_{ij}$: scalar / z_{ij} : vector
  - z_{ij}와 같은 크기의 vector w^*를 $\hat{x}_{ij}$에 곱 (w^*는 learnable)
  - The updated embedding $\tilde{z}_{ij}$

$$\tilde{\mathbf{z}}_{ij} = \begin{cases} Att(\tilde{\mathbf{x}}_{ij} \cdot \mathbf{w}^*, \mathbf{z}_{ij}) & e_j \in \bigcup_k S_i^k \\ \hat{\mathbf{x}}_{ij} \cdot \mathbf{w}^* & otherwise \end{cases}$$



where $\tilde{\mathbf{x}}_{ij} = \frac{\hat{\mathbf{x}}_{ij}}{\max_j \hat{\mathbf{x}}_{ij}} - 0.5$ which normalizes all values of $\hat{\mathbf{x}}_i$ to $[0.5, -0.5]$

$$Att(\mathbf{q}, \mathbf{k}) = \alpha \cdot \mathbf{q} + (1-\alpha) \cdot \mathbf{k}, \text{ where } \alpha = \sigma\left((\mathbf{W}_q \cdot \mathbf{q})^T (\mathbf{W}_k \cdot \mathbf{k})\right)$$

$$\hat{\mathbf{y}}_{ij} = \sigma\left(\tilde{\mathbf{z}}_{ij}^T \mathbf{w}_0 + b_0\right) \quad \sigma(x) = \frac{1}{1+e^{-x}}$$

$$\mathcal{L}_{TSP} = -\frac{1}{N} \sum_i^N \sum_j^M \mathbf{y}_{ij} \log \hat{\mathbf{y}}_{ij} + (1 - \mathbf{y}_{ij}) \log(1 - \hat{\mathbf{y}}_{ij})$$

Global-Local Item Embedding for Temporal Set Prediction

- **modeling**
  - VAE, DNNTSP 30 epochs으로 훈련, VAE은 1 layer 사용
  - τ = 0.6가 가장 좋은 결과를 보임
  - latent space의 차원은 DC, TaoBao : 128 / TMS : 512
  - Adam optimizer / learning rate : 0.001

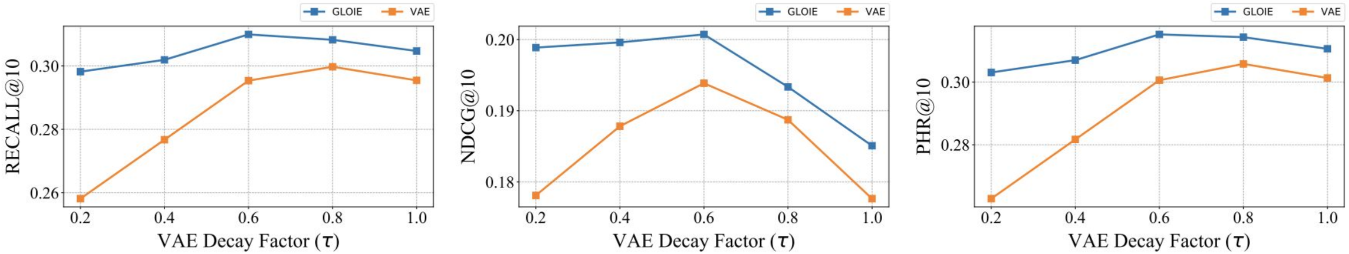


Fig. 3. Metric@10 by varying VAE decay factor (τ) on TaoBao dataset.

Global-Local Item Embedding for Temporal Set Prediction

- DC/Taobao dataset
    - 모든 metric에서 가장 높음
- top K = 10
    - 모든 metric에서 DNNTSP 능가
- the number of items a user interacted
    - DC : 5.44 / TaoBao : 4.96, TMS : 18.05
- VAE decoder의 output distribution이 성능에 크게 영향을 줌

Table 2. Comparison between various state-of-the-art methods and ours on three public benchmarks. All highest scores are in **bold** and all second best scores are underlined.

Dataset	Model	k = 10			k = 20			k = 40		
		Recall	NDCG	PHR	Recall	NDCG	PHR	Recall	NDCG	PHR
DC	Toppop	0.1618	0.0880	0.2274	0.2475	0.1116	0.3289	0.3940	0.1448	0.4997
	PersonalToppop	0.4104	0.3174	0.5031	0.4293	0.3270	0.5258	0.4747	0.3332	0.5785
	Sets2Sets	0.4488	0.3136	0.5458	0.5143	0.3319	0.6162	0.6017	0.3516	0.7005
	DNNTSP	0.4564	0.3165	0.5557	0.5294	0.3369	0.6272	0.6180	0.3568	0.7165
	VAE - *Gaussian*	0.1618	0.0882	0.2274	0.2507	0.1128	0.3333	0.3847	0.1430	0.4903
	VAE - *Multinomial*	0.1602	0.0850	0.2230	0.2492	0.1097	0.3311	0.3767	0.1387	0.4786
	VAE - *Tweedie*	0.4166	0.3000	0.5108	0.5122	0.3267	0.6062	0.6217	0.3517	0.7088
	GLOIE - *Gaussian*	0.3108	0.2349	0.3971	0.3738	0.2526	0.4664	0.4545	0.2706	0.5563
	GLOIE - *Multinomial*	0.3265	0.2465	0.4143	0.3870	0.2633	0.4798	0.4615	0.2803	0.5602
	GLOIE - *Tweedie*	**0.4658**	**0.3264**	**0.5613**	**0.5415**	**0.3477**	**0.6351**	**0.6428**	**0.3708**	**0.7288**
TaoBao	Toppop	0.1567	0.0784	0.1613	0.2494	0.1019	0.2545	0.3679	0.1264	0.3745
	PersonalToppop	0.2190	0.1535	0.2230	0.2260	0.1554	0.2306	0.2433	0.1590	0.2484
	Sets2Sets	0.2811	0.1495	0.2868	0.3649	0.1710	0.3713	0.4672	0.1922	0.4739
	DNNTSP	0.3035	0.1841	0.3095	0.3811	0.2039	0.3873	0.4776	0.2238	0.4843
	VAE - *Gaussian*	0.1592	0.0750	0.1635	0.2480	0.0974	0.2530	0.3665	0.1219	0.3727
	VAE - *Multinomial*	0.1588	0.0798	0.1634	0.2494	0.1027	0.2545	0.3660	0.1268	0.3723
	VAE - *Tweedie*	0.2954	0.1939	0.3006	0.3775	0.2148	0.3827	0.4768	0.2353	0.4822
	GLOIE - *Gaussian*	0.2982	0.1768	0.3044	0.3790	0.1973	0.3851	0.4769	0.2175	0.4835
	GLOIE - *Multinomial*	0.2980	0.1791	0.3040	0.3783	0.1995	0.3846	0.4750	0.2195	0.4819
	GLOIE - *Tweedie*	**0.3099**	**0.2007**	**0.3152**	**0.3917**	**0.2216**	**0.3972**	**0.4868**	**0.2412**	**0.4924**
TMS	Toppop	0.2627	0.1627	0.4619	0.3902	0.2017	0.6243	0.5605	0.2448	0.8007
	PersonalToppop	0.4508	0.3464	0.6440	0.5274	0.3721	0.7146	0.5495	0.3771	0.7374
	Sets2Sets	0.4748	0.3782	**0.6933**	0.5601	0.4061	0.7594	0.6627	0.4321	0.8570
	DNNTSP	0.4693	0.3473	0.6825	0.5826	0.3839	**0.7880**	0.6840	0.4097	**0.8748**
	VAE - *Gaussian*	0.2731	0.1919	0.4660	0.3913	0.2288	0.6195	0.5496	0.2688	0.7813
	VAE - *Multinomial*	0.2548	0.1615	0.4431	0.3830	0.2001	0.6020	0.5437	0.2412	0.7740
	VAE - *Tweedie*	0.4661	0.3744	0.6548	0.5579	0.4040	0.7432	0.6663	0.4316	0.8341
	GLOIE - *Gaussian*	0.1345	0.0833	0.2486	0.2363	0.1155	0.4018	0.4014	0.1570	0.6033
	GLOIE - *Multinomial*	0.1479	0.1029	0.2797	0.2192	0.1252	0.3872	0.3259	0.1524	0.5362
	GLOIE - *Tweedie*	**0.4860**	**0.3823**	0.6863	**0.5868**	**0.4144**	0.7753	**0.6926**	**0.4418**	0.8538

Q&A