Knowledge Enhanced Multi-Interest Network for the Generation of Recommendation Candidates

Danyang Liu*
ldy591@mail.ustc.edu.cn
University of Science and Technology of China & Meituan
Hefei, China

Yuji Yang*
Meituan
Beijing, China
yangyuji02@meituan.com

Mengdi Zhang
Meituan
Beijing, China
zhangmengdi02@meituan.com

Wei Wu
Meituan
Beijing, China
wuwei19850318@gmail.com

Xing Xie
Microsoft Research
Beijing, China
xingx@microsoft.com

Guangzhong Sun
University of Science and Technology of China
Hefei, China
gzsun@ustc.edu.cn

https://github.com/danyang-liu/KEMI

— CIKM 2022

Reported by Yabo Yin

1.Introduction

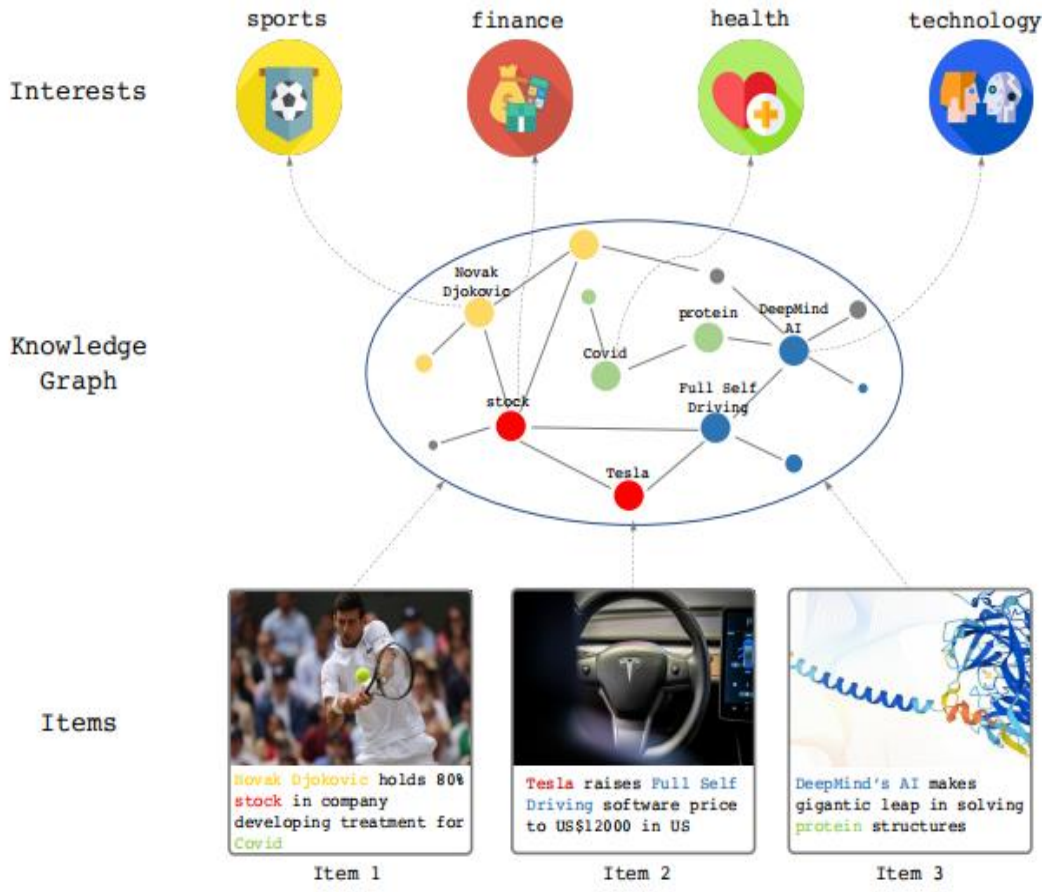
2.Method

3.Experiments

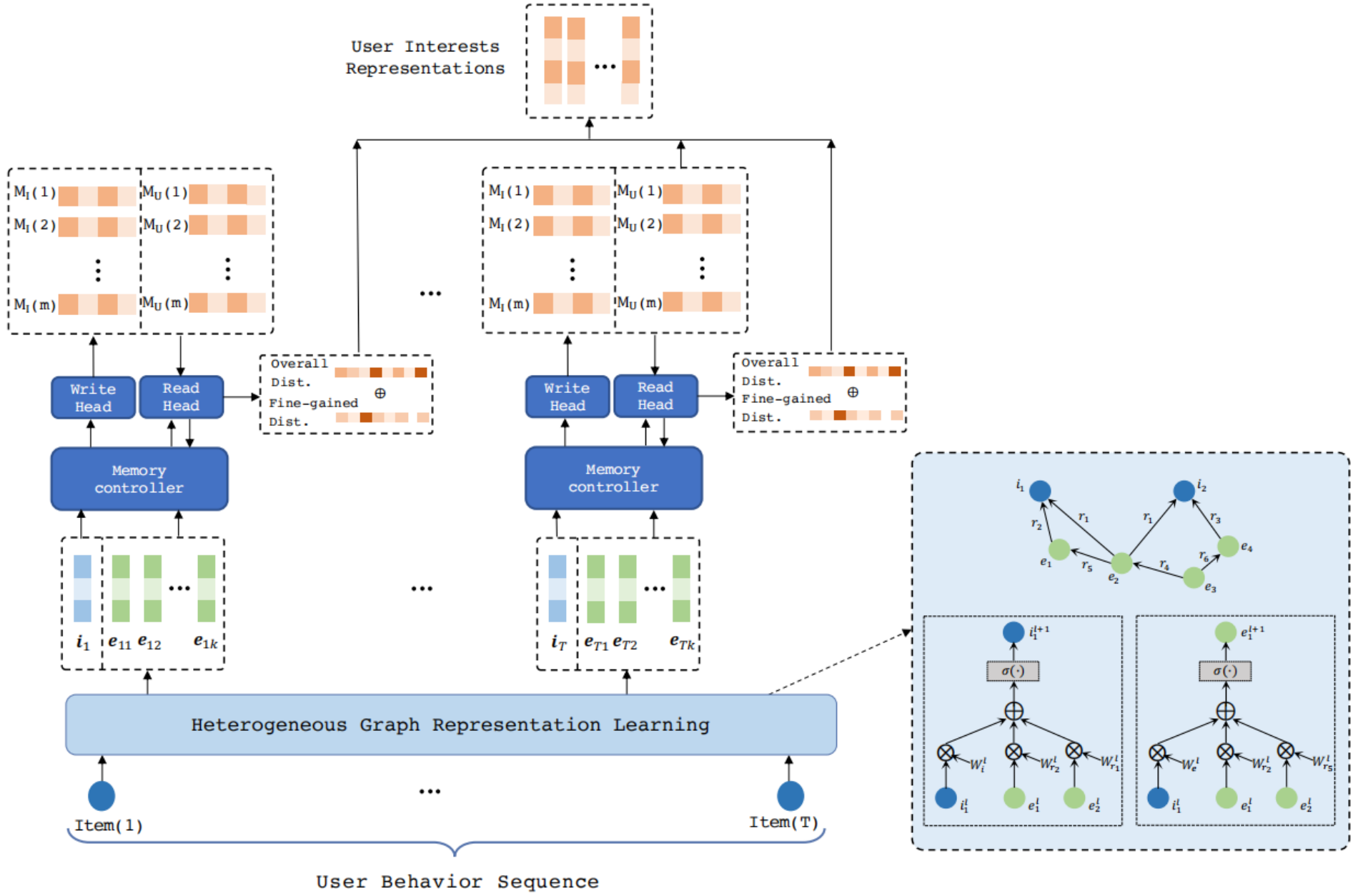
Introduction

1. Previous methods cluster users' behavior sequences into different groups to represent different interests, ignoring that an item may contain multiple fine-grained interests.

2. Explaining user interests at the level of items is rather vague and not convincing.



Method


User Interests Representations
Write Head
Read Head
Memory controller
Overall Dist.
Fine-gained Dist.
Heterogeneous Graph Representation Learning
Item(1)
Item(T)
User Behavior Sequence

Method

Preliminaries

user historical behaviors $H_u = \{i_1^u, i_2^u, ..., i_T^u\}$

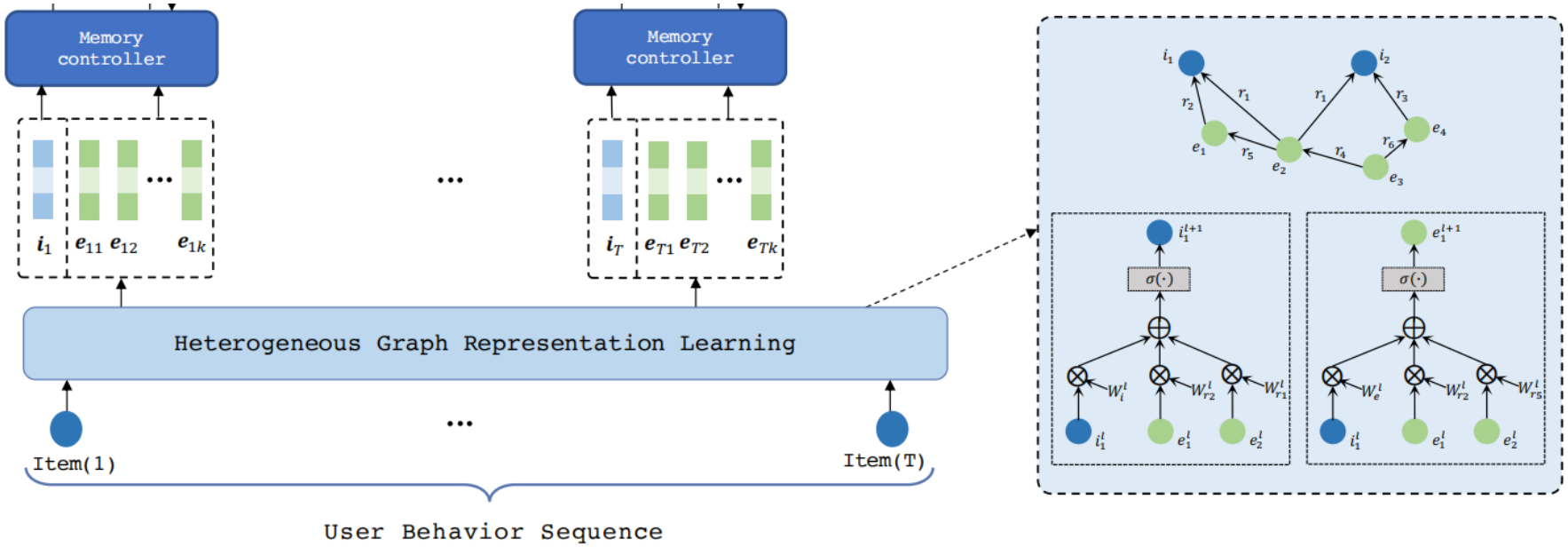
Each item $i \in \mathcal{I}$ $E_i = \{e_1^i, e_2^i, ..., e_k^i\}$.

The core task

$$\mathbf{V}_u = f_{user}(H_u) \quad (1)$$

where $\mathbf{V}_u = \{\mathbf{v}_1^u, \mathbf{v}_2^u, ..., \mathbf{v}_m^u\} \in \mathbb{R}^{m \times d}$,

Method



Graph Representation Learning

$$\mathbf{h}_v = JK(\mathbf{h}_v^0, ..., \mathbf{h}_v^n) \quad (2)$$

$$\mathbf{h}_v^{l+1} = U_l(\mathbf{h}_v^l, \mathbf{m}_v^{l+1}) \quad (3)$$

$$\mathbf{h}_v^0 = \begin{cases} \mathbf{h}_v^{0T} & v \in entities \\ concat(\mathbf{h}_v^{0B}, \mathbf{h}_v^{0T}) & v \in items \end{cases} \quad (4)$$

$$\mathbf{m}_v^{l+1} = \sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^r} \frac{1}{c_{i,r}} \mathbf{W}_r^{(l)} \mathbf{h}_j^{(l)} \quad (5)$$

where $\mathcal{N}_i^r$ denotes the set of neighbor indices of node i under relation $r \in R$. $c_{i,r}$ is a problem-specific normalization constant.

$$\mathbf{h}_v^{l+1} = \sigma\left(\mathbf{m}_v^{l+1} + \mathbf{W}_0^l \mathbf{h}_i^l\right) \quad (6)$$

Method

In the time step of t, parameter of memory is indicated as M_t , which consists of m memory slots, representing m different interests.

At time step t, we have $(\mathbf{i}_t, \{\mathbf{e}_t^1, \mathbf{e}_t^2, ...\mathbf{e}_t^k\})$,

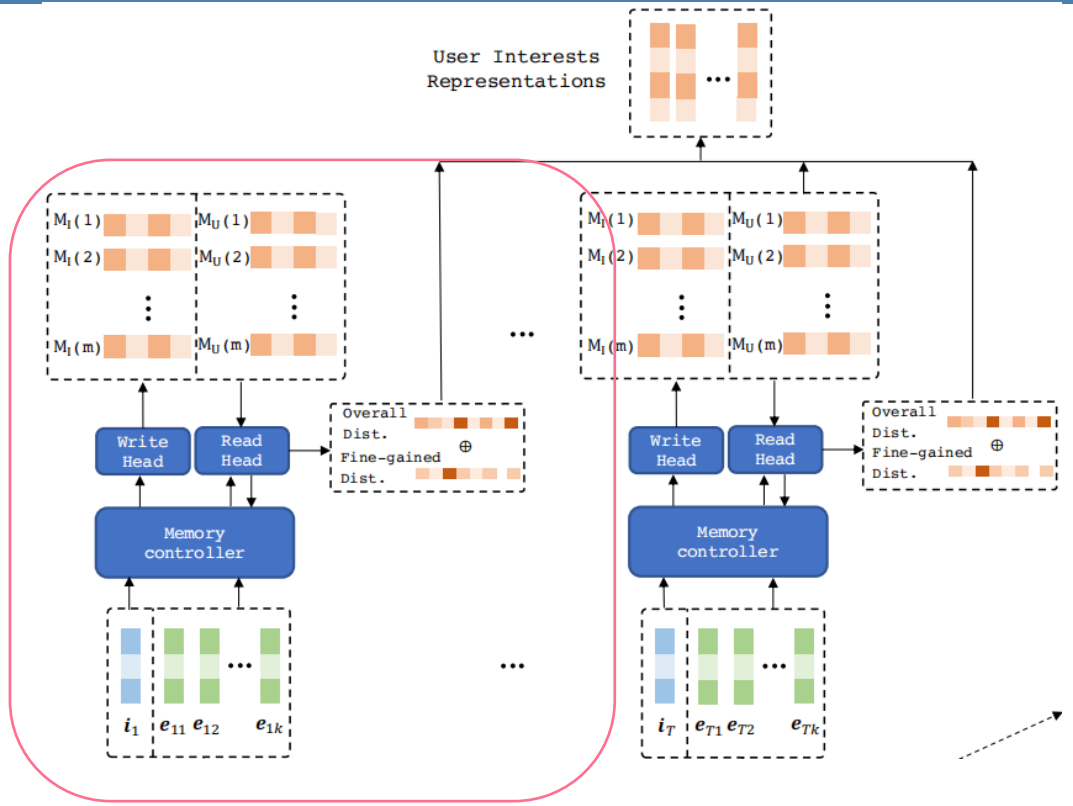
Read Interest Distributions:

$$\mathbf{k}_i^t = \mathbf{i}_t \mathbf{K} \tag{7}$$

$$\mathbf{w}_l^t(i) = \frac{exp(f(\mathbf{k}_i^t, M_I^t(l)))}{\sum_j^m exp(f(\mathbf{k}_i^t, M_I^t(j)))}, \quad for\ l = 1, 2, ..., m \tag{8}$$

where,

$$f(\mathbf{k}_i^t, M_I^t(j)) = \frac{\mathbf{k}_i^{tT} M_I^t(j)}{\|\mathbf{k}_i^t\| \|M_I^t(j)\|} \tag{9}$$



$$\mathbf{k}_e^t = (\sum_j^k \mathbf{e}_j^t)\mathbf{K} \tag{10}$$

$$\mathbf{w}_l^t(e) = \frac{exp(f(\mathbf{k}_e^t, M_I^t(l)))}{\sum_j^m exp(f(\mathbf{k}_e^t, M_I^t(j)))}, \quad for\ l = 1, 2, ..., m \tag{11}$$

where,

$$f(\mathbf{k}_e^t, M_I^t(j)) = \frac{\mathbf{k}_e^{tT} M_I^t(j)}{\|\mathbf{k}_e^t\| \|M_I^t(j)\|} \tag{12}$$

The interest distribution for item i_t for user u is:

$$\mathbf{w}_u^t = \alpha * \mathbf{w}^t(i) + (1 - \alpha) * \mathbf{w}^t(e) \tag{13}$$

Method

$$\mathbf{a}_i = (\sum_{t=1}^{T} \mathbf{i}_t + \sum_{t=1}^{T}\sum_{j=1}^{k} \mathbf{e}_t^j)\mathbf{A} \tag{14}$$

$$\mathbf{e}_i = (\sum_{t=1}^{T} \mathbf{i}_t + \sum_{t=1}^{T}\sum_{j=1}^{k} \mathbf{e}_t^j)\mathbf{E} \tag{15}$$

$$\mathbf{M}_I = (1 - \mathbf{E}_I) \odot \mathbf{M}_I + \mathbf{A}_I \tag{16}$$

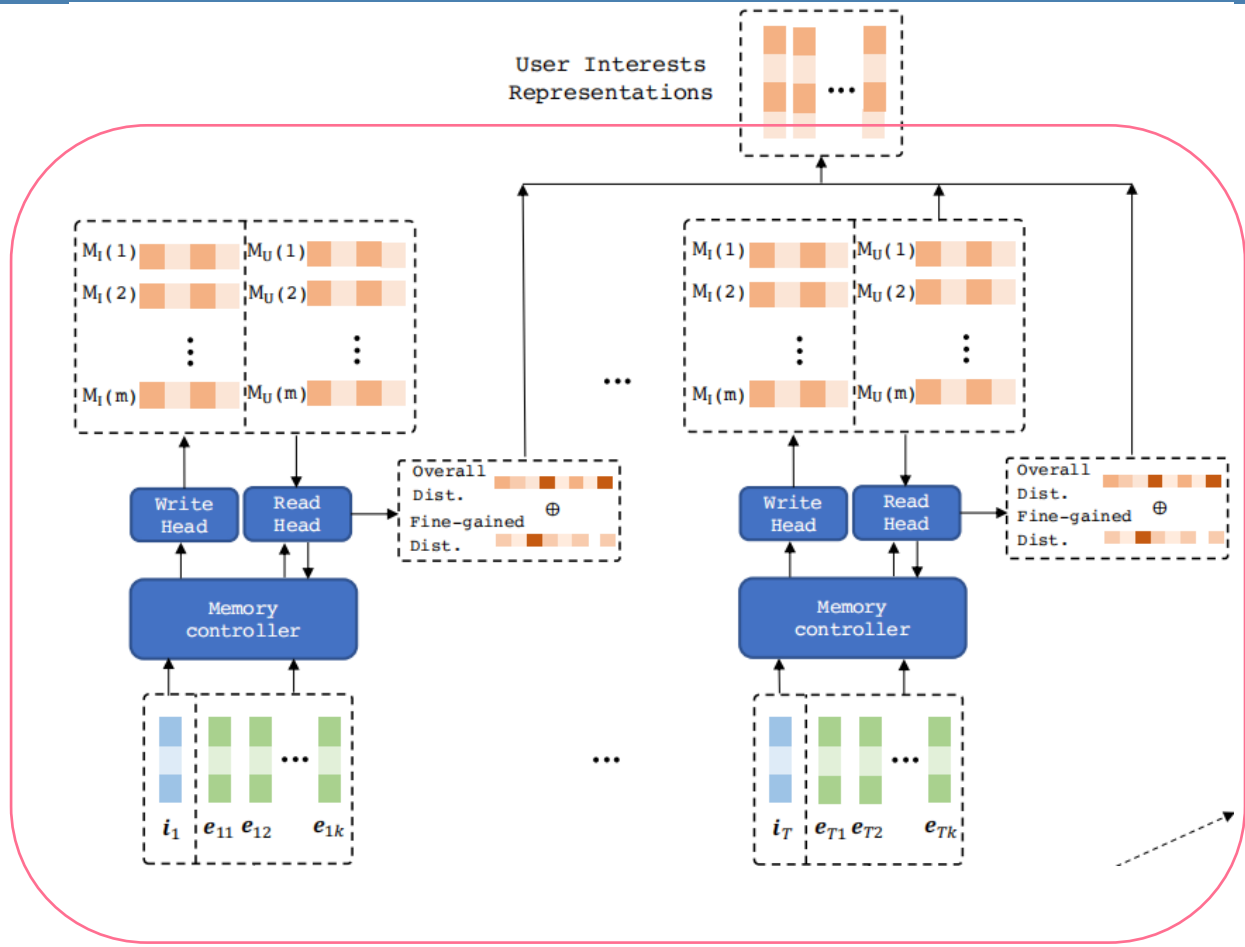
where $\mathbf{A}$ is add matrix in controller, $\mathbf{E}$ is erase matrix in controller, $\mathbf{E}_I = \mathbf{w}_u \otimes \mathbf{e}_i$, $\mathbf{A}_I = \mathbf{w}_u \otimes \mathbf{a}_i$, and $\odot$ and $\otimes$ means dot product and outer product respectively.

$$\mathbf{a}_u^t = (\mathbf{i}_t + \sum_{j=1}^{k} \mathbf{e}_t^j)\mathbf{A} \tag{17}$$

$$\mathbf{e}_u^t = (\mathbf{i}_t + \sum_{j=1}^{k} \mathbf{e}_t^j)\mathbf{E} \tag{18}$$

$$\mathbf{M}_U^t = (1 - \mathbf{E}_U^t) \odot \mathbf{M}_U^{t-1} + \mathbf{A}_U^t \tag{19}$$

where $\mathbf{E}_U^t = \mathbf{w}_u^t \otimes \mathbf{e}_u^t$ and $\mathbf{A}_U^t = \mathbf{w}_u^t \otimes \mathbf{a}_u^t$.



Method

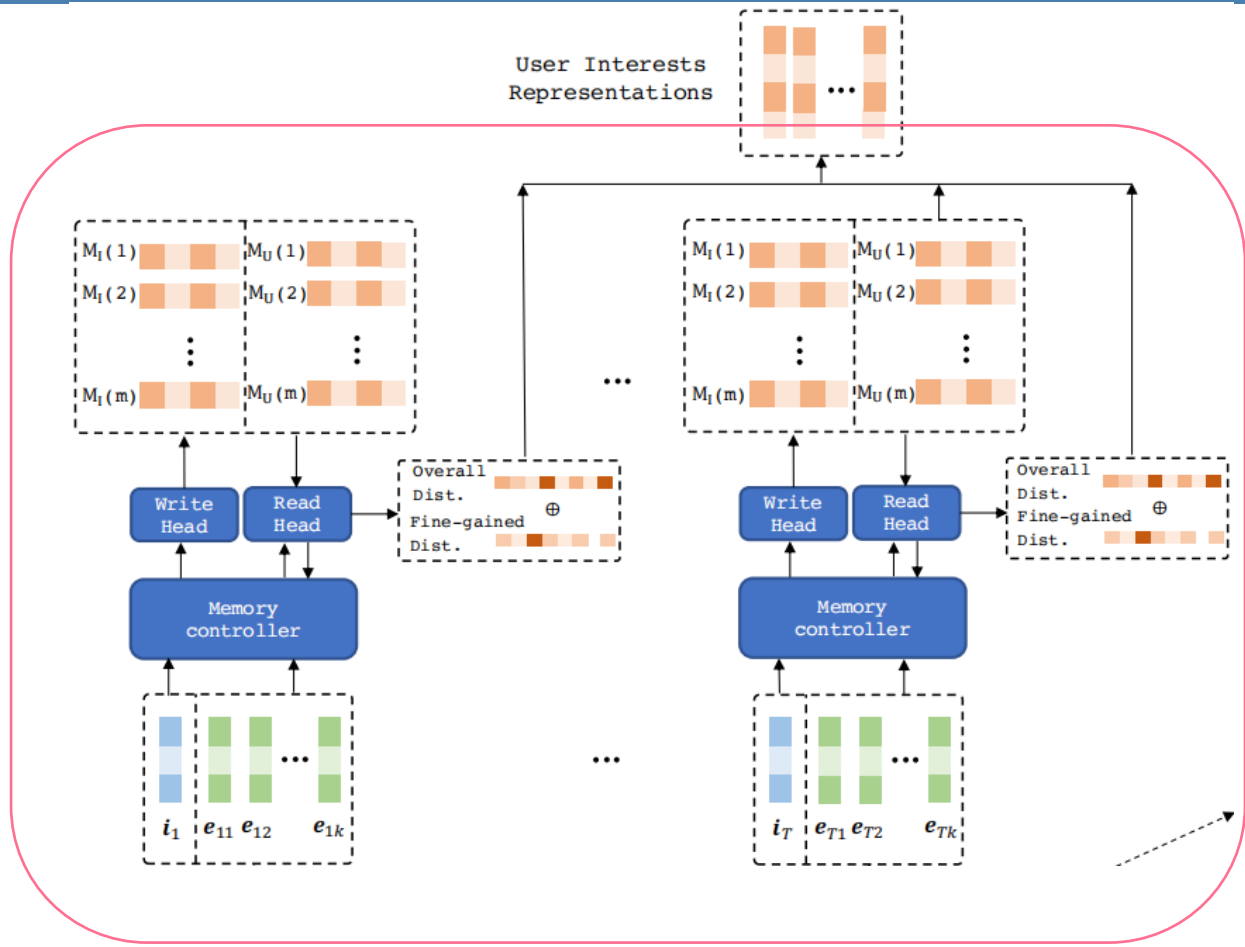
$$\mathbf{V}_u = (\sum_{t=1}^{T} \mathbf{w}_u^t)\mathbf{M}_U^T \tag{20}$$

Learning

$$\mathcal{L}_i = KL(\mathbf{w}(i)|\mathbf{w}(e)) = \sum_{j=1}^{m}\sum_{t=1}^{T} w(i)_j^t log\frac{w(i)_j^t}{w(e)_j^t} \tag{21}$$

$$\mathcal{L}_d = dist(\mathbf{v}_u^k, \mathbf{i}^k) \tag{22}$$

$$\mathcal{L} = \mathcal{L}_d + \lambda_1\mathcal{L}_i + \lambda_2 \|\Phi\|_2^2 \tag{23}$$



Experiments

	Microsoft News	Dianping Feed
# . users	1,000,000	100,000
# . items	161,013	915,493
#. interactions	24,155,470	1,518,490
#. associated entities per item	17.2	15.8
#. words per article	639	137

Table 2: Statistics of the two realistic datasets.

	Wikidata	Private KG
#. entities	3,275,149	2,502,554
#. triples	31,963,632	19,467,000
#. relations	1,091	47

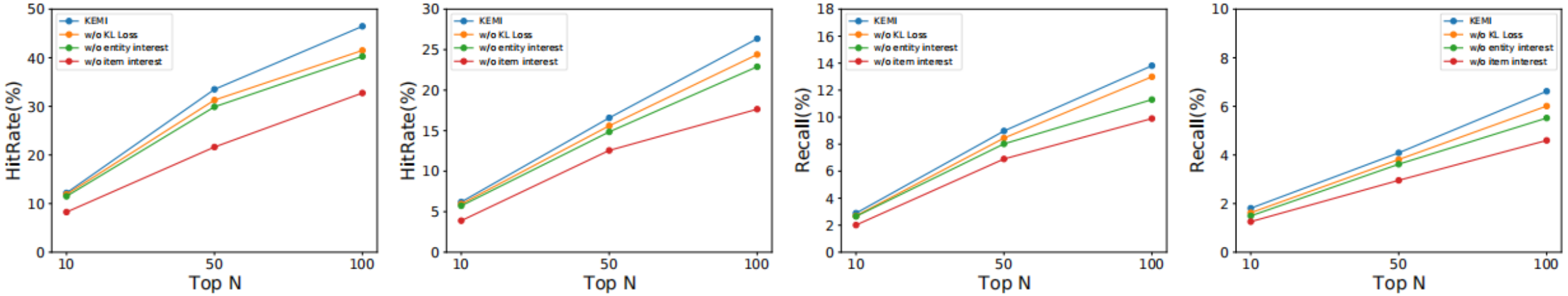
Table 3: Statistics of the knowledge graphs.

Experiments

	Dianping Feed Dataset						Microsoft News Dataset					
	Metrics@10		Metrics@50		Metrics@100		Metrics@10		Metrics@50		Metrics@100	
	Hit Rate	Recall	Hit Rate	Recall	Hit Rate	Recall	Hit Rate	Recall	Hit Rate	Recall	Hit Rate	Recall
Most Popular	5.713	1.014	15.74	3.215	23.28	5.385	1.659	0.526	1.696	0.536	1.704	0.537
Youtube DNN	11.36	2.586	30.69	8.081	41.54	12.44	5.832	1.492	14.87	3.814	22.29	6.079
MIND	11.81	2.830	31.89	8.900	43.19	13.39	3.113	0.752	12.77	3.318	20.93	5.728
Comirec	11.44	2.742	31.47	8.905	42.27	13.28	3.753	0.941	13.57	3.489	22.22	6.160
Octopus	11.49	2.768	31.99	8.913	42.89	13.12	5.845	1.543	15.60	3.616	23.90	6.198
MIMN	11.28	2.601	30.98	8.426	42.35	12.90	5.032	1.454	14.31	3.732	21.50	5.833
KEMI	**12.19**	**2.889**	**33.50**	**8.976**	**46.47**	**13.81**	**6.198**	**1.808**	**16.58**	**4.088**	**26.35**	**6.621**

Table 4: Model performance on two datasets. Bolded numbers are the best performance of each column. The results are reported in percentage (%).

Experiments



(a) HitRate on Feed Dataset (b) HitRate on News Dataset (c) Recall on Feed Dataset (d) Recall on News Dataset

Figure 3: Ablation study on removing the constrained interest loss, the overall interest weight or the fine-grained interest weight on two datasets.

Experiments

Interests	News Title	Knowledge Entity
Interest 1	News1: Lloyd: Coming off bench at World Cup "rock bottom of my entire career" News2: Premier League player Power Rankings News3: Lukaku brace at Bologna fires Inter top, Roma beat Napoli to go third	Premier League Cristiano Ronaldo World Cup FC Barcelona
Interest 2	News1: Amazon CEO Jeff Bezos is no longer the richest person in the world News2: Starbucks gives $10 million to help boost small business in Chicago's neighborhoods News3: McDermott chief financial officer resigns following $1.9 billion loss	CFO stock SoftBank credict card
Interest 3	News1: Why macOS Catalina is breaking so many apps, and what to do about it News2: Pixel 4 and 4 XL hands-on: Google ditches fingerprint scanner for face unlock News3: Xbox's latest safety feature lets you filter out offensive trash talk	Xbox macOS Catalina Pixel 4 Apple

Figure 4: An example of news and knowledge entities related to the different interest channels of our KEMI model

		Microsoft News			Dianping Feed		
		N=10	N=50	N=100	N=10	N=50	N=100
m=1	HR@N	5.806	14.13	22.43	11.21	30.40	41.96
	Recall@N	1.518	3.725	6.002	2.667	8.142	12.48
m=5	HR@N	5.892	14.99	23.20	11.83	32.19	44.80
	Recall@N	1.653	3.816	6.190	2.878	8.951	13.38
m=10	HR@N	6.062	15.81	25.47	**12.19**	**33.50**	**46.47**
	Recall@N	1.775	3.901	6.472	**2.889**	**8.976**	**13.81**
m=20	HR@N	**6.198**	**16.58**	26.35	12.02	33.10	46.01
	Recall@N	**1.808**	**4.088**	**6.621**	2.841	8.953	13.46
m=30	HR@N	6.134	16.46	**26.48**	11.60	32.18	44.63
	Recall@N	1.769	3.957	6.608	2.841	8.880	43.69

Table 5: Model performance w.r.t. different number of interests, the results are reported in percentage (%). Bolded numbers are the best performance.

Experiments

# users	# items	#interactions
2,659,562	2,123,510	160,409,918

Table 6: Statistics of the industrial large dataset

sults demonstrate that our KEMI model can improve Recall@100 and HR@100 with 3.73% and 6.25% compared to the best baseline results respectively.

Experiments

Our training process is formalized as Alg. 1

Algorithm 1: Training Procedure of KEMI

Input: user behavior sequence H_u; knowledge graph $\mathcal{G}$;
Output: user representation function f_{user};

Randomly initialize all parameters ;
while *not converge* **do**
  for *each user u in mini-batch* U_i **do**
    for *item i at time step t* **do**
      Learn unified representations of item and entities: $(\mathbf{i}_t, \{\mathbf{e}_t^1, \mathbf{e}_t^2, ...\mathbf{e}_t^k\})$. ▷ Section 3.2 ;
      Read interest memory network M_I, get overall and fine-grained interest distributions: $\mathbf{w}^t(i)$ and $\mathbf{w}^t(e)$. ▷ Section 3.3.1 ;
      Write User Memory Network M_U. ▷ Section 3.3.2 ;
    Write Interest Memory Network M_I. ▷ Section 3.3.1;
    Get u's representation $\mathbf{V}_u$. ▷ Section 3.3.3 ;
    Calculate overall objective function $\mathcal{L}$. ▷ Section 3.4 ;
  Back-propagate the gradients and update.

Thank you!