Path-enhanced Explainable Recommendation with Knowledge Graphs

Yafan Huang, Feng Zhao†, Shihui Song, Xiangyu Gui, Hai Jin

Huazhong University of Science and Technology, Wuhan, China

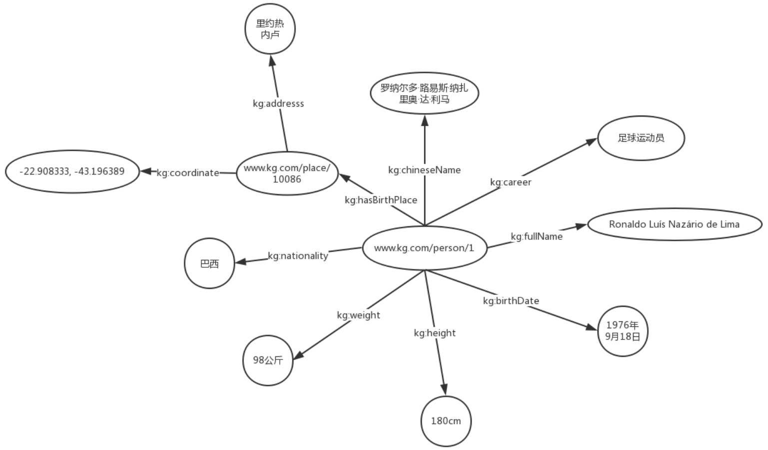
May 13, 2020

- **Recommender system** helps us to obtain information/products that interest us more.

- A **Triple** combines two entities with one relation, e.g. :

$$(Michael\ Jackson, Born\_in, Indiana)$$

- **Knowledge Graph (KG)**, composed of countless triples, is a well-structured data form and has already used in recommender system, question answering, etc.

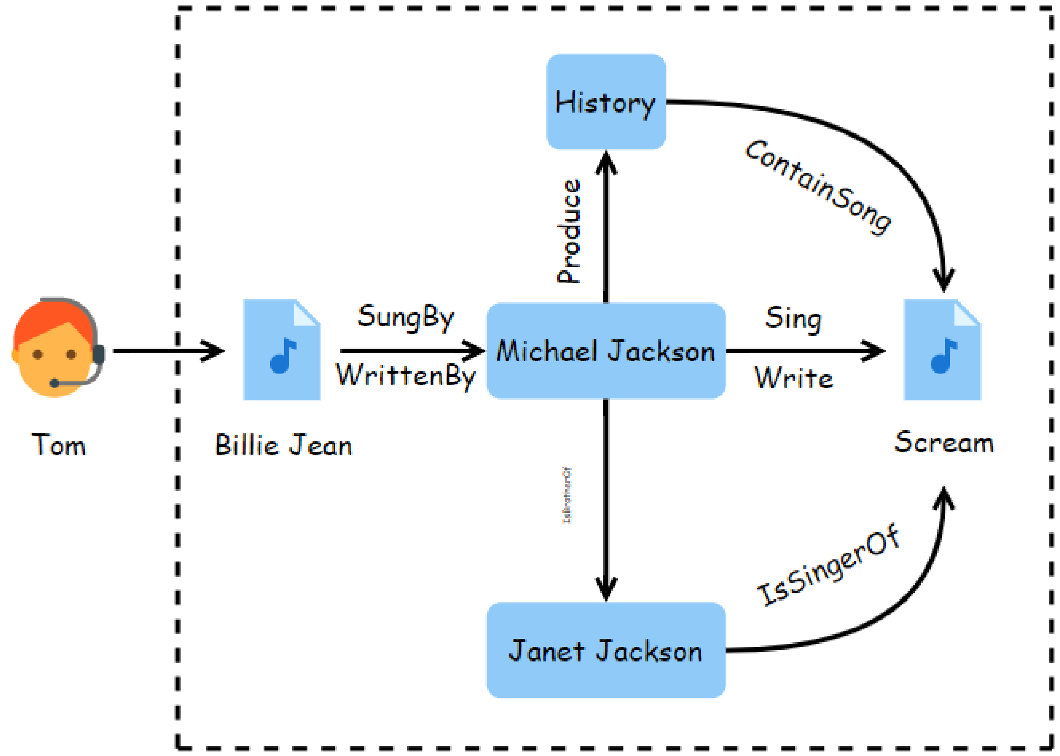


And what will happen with:

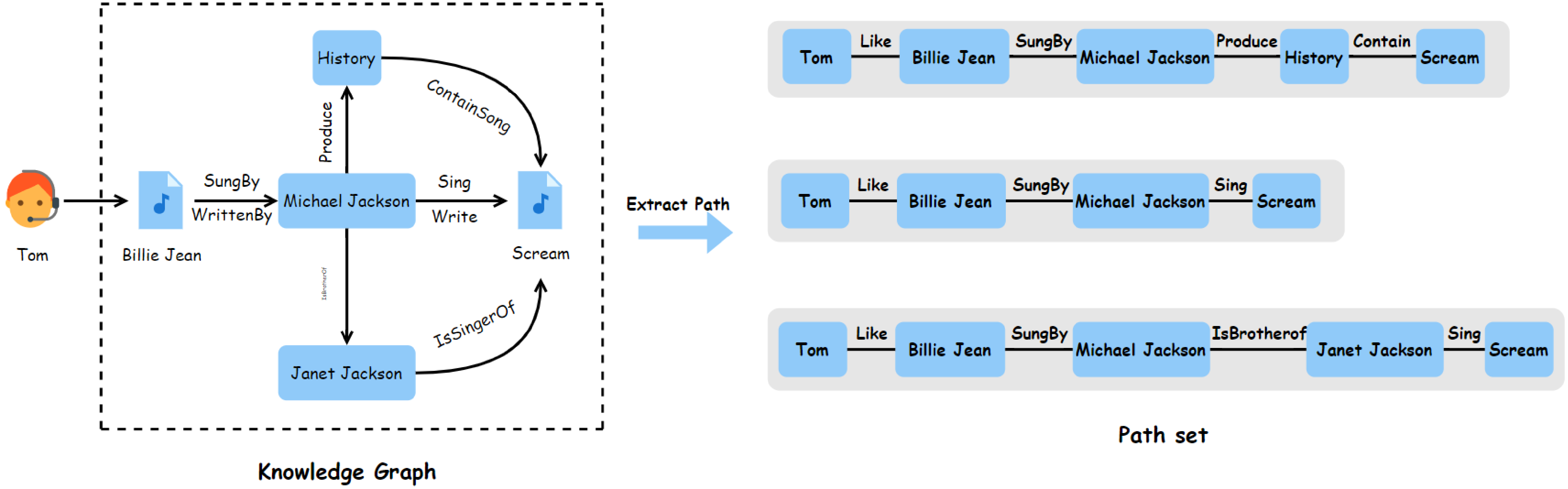
Recommender system + Knowledge Graph?

We give a toy example:

- One day, Tom listened to song “Billie Jean”.
- Then, Tom was recommended a song “Scream”.
- Tom did like “Scream”!
- Why?
- To explain this, we draw a music field Knowledge Graph, as shown in right side:



Knowledge Graph

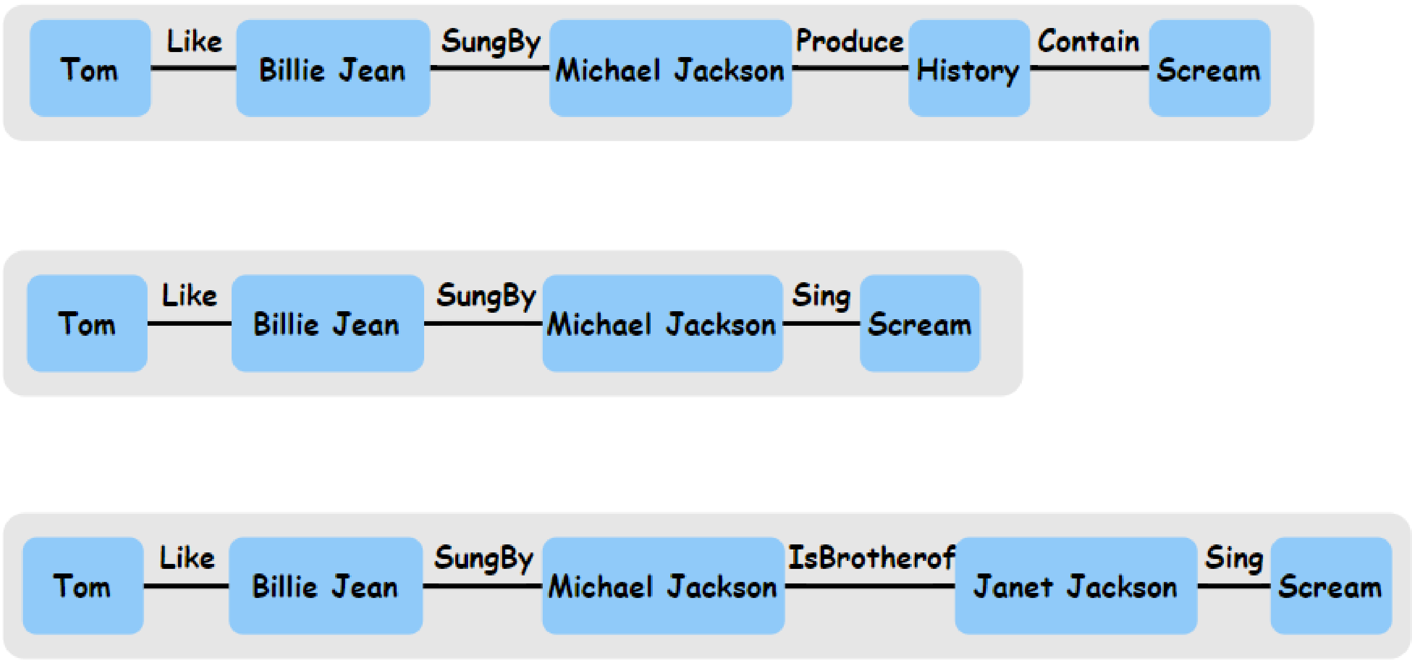


- Three extracted paths explain the recommend reason.
- **Explainability** is significantly enhanced compared with Collaborative Filtering (similar users tend to like similar items).

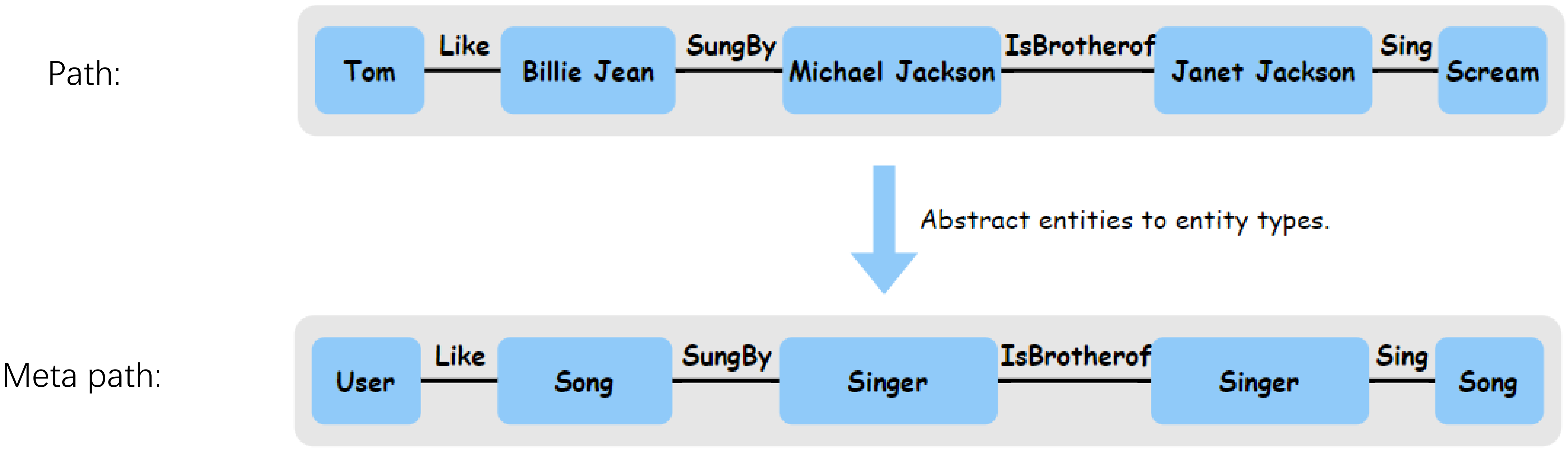
However, there are still two questions:

- Q1: How to extract paths from Knowledge Graph quickly and accurately?

- Q2. Which path here accounts for “Tom likes Scream”more?



To solve these, we introduce **meta path**.



- A Meta Path can represent a series of paths with **the same structure**.

With **meta path**, here comes our solutions:

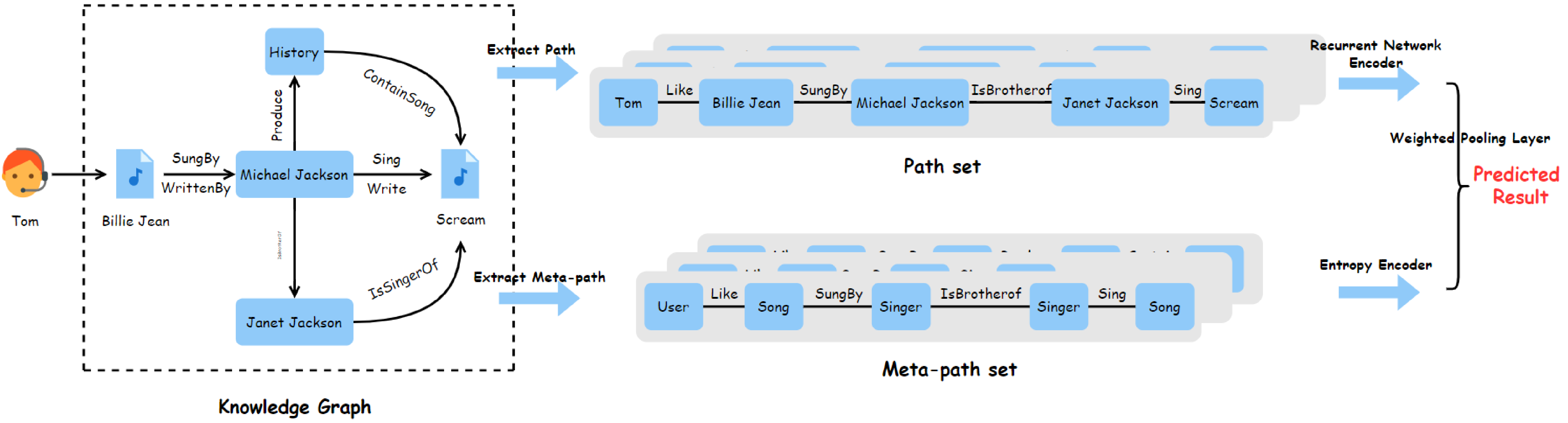
- Q1: How to extract paths from Knowledge Graph quickly and accurately?

Meta-path-aided
Path Extraction Algorithm

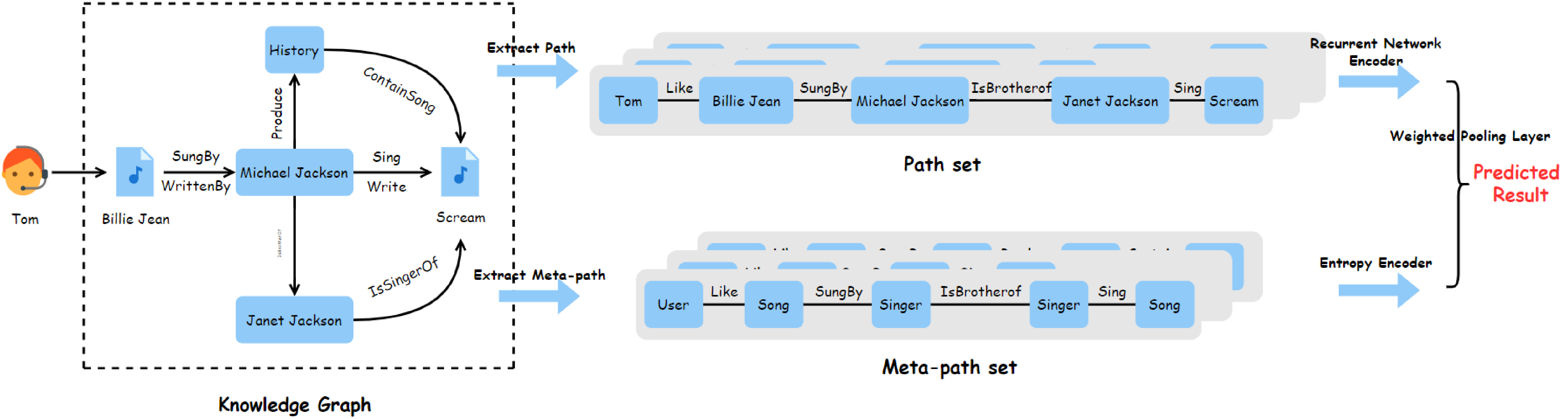
- Q2. Which path here accounts for “Tom likes Scream”more?

Meta-path-based
Entropy Encoder

Overall framework of our proposed model:



Name: **P**ath-**e**nhanced **R**ecurrent **N**etwork (PeRN)



Some explanations of this model:

- PeRN is an end-to-end recommendation model.
- Path Extraction Algorithm is used in "Extract Path" step.
- Meta path is abstracted from path.
- $(Tom, Scream)$ here is defined as an user-item interaction.

Methodology

- Path Extraction Algorithm
- Recurrent Network Encoder
- Entropy Encoder
- Weighted Pooling Layer
- Optimization

Path Extraction Algorithm

Some explanations:

- This algorithm is meta-path-aided.
- This algorithm mainly uses the idea of bi-directional search.
- We regard paths over six hops as noise.
- **if $\boldsymbol{p_l\ satisfies\ mp}$** indicates the condition when meta path of p_l is same as mp.

Algorithm 1: Bidirectional Path Extraction

Input: Knowledge graph $\mathcal{KG} = \{(h, r, t) | h, t \in \mathcal{E}, r \in \mathcal{R}\}$,
user-item interaction set $\mathcal{A} = \{a_1, a_2, \dots a_{|\mathcal{A}|}\}$,
meta path set $\mathcal{M} = \{mp_1, mp_2, \dots, mp_{|\mathcal{M}|}\}$.

Output: Path set $\mathcal{P} = \{P_1, P_2, \dots, P_{|\mathcal{A}|}\}$.

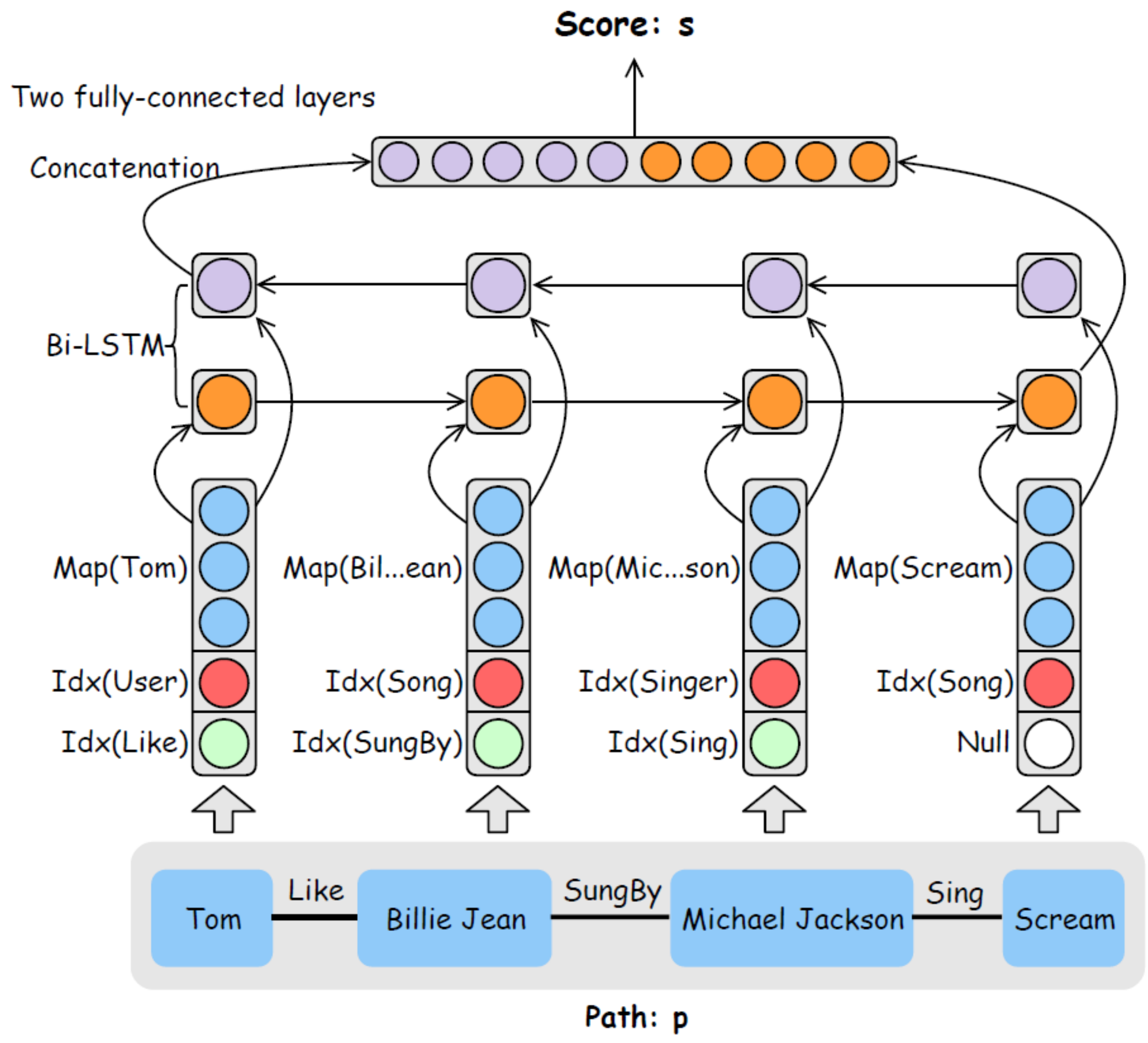
Initialize: $\mathcal{P} \leftarrow \varnothing$;
for each $a_k = (u_m, i_n)$ *in* $\mathcal{A}$ **do**
    $P_k \leftarrow \varnothing$;
    $S_1 \leftarrow$ retrieve all paths by head = u_m within 3 hops;
    $S_2 \leftarrow$ retrieve all paths by head = i_n within 3 hops;
    for each *meta path mp in* $\mathcal{M}$ **do**
        $l \leftarrow$ length of mp;
        $P_1 \leftarrow \varnothing$ // left sub path set;
        $P_2 \leftarrow \varnothing$ // right sub path set;
        for each p_l *in* S_1 **do**
            if p_l *satisfies* $mp[0 \rightarrow 2 * \lfloor l/2 \rfloor]$ **then**
                Add p_l to P_1;
        for each p_r *in* S_2 **do**
            if p_r *satisfies* $mp[2 * \lfloor l/2 \rfloor \rightarrow 2 * l]$ **then**
                Add p_r to P_2;
        for each p_l, p_r *in* P_1, P_2 **do**
            if $p_l[tail] = p_r[tail]$ **then**
                $p \leftarrow$ combine p_l and reverse(p_r);
                Add p to P_k;
    Add P_k to $\mathcal{P}$;
return $\mathcal{P}$

Recurrent Network Encoder

Some explanations:

- An n-hop path is composed of n triples.
- The format of embedded vector is $(entity, \text{typeof}(entity), relation)$, so an n-hop path will be embedded to n+1 vectors.
- Two fully-connected layers and activation function:

$$s = W_2^T \text{ReLU}(W_1^T h)$$



Entropy Encoder

$$w_i = \frac{\mathrm{Gain}(D, E_g = mp_i)}{\sum_{j=1}^{|MP\_k|} \mathrm{Gain}(D, E_g = mp_j)}$$

Some explanations:

- w_i is designed for weighting different paths in one user-item interaction.
- Each path in one user-item interaction holds its own weight w.
- In one user-item interaction $a_k = (u,\ i)$, there might be n paths which can be abstracted to m meta paths ($n \geq m$). The type of meta path can be seen as a feature in a_k.
- $\mathrm{Gain}(D, E_g = mp_i)$ indicates the **information gain** from feature E_g (the type of meta path) to D (if this path is “right”) .

Weighted Pooling Layer

$$\widehat{y_k} = \sigma(\sum_{i=1}^{|P_k|} w_i\, s_i)$$

Some explanations:

- P_k is the path set of user-item interaction a_k.
- $\boldsymbol{w_i}$ and $\boldsymbol{s_i}$ is the weight and score of i-th path in P_k.
- σ indicates sigmoid activation function.

Optimization

$$L = -\sum_{a \in A} (y \log \hat{y} + (1 - y) \log(1 - \hat{y}))$$

L_2 regularization is conducted here, which is omitted for simplicity.

Experiments

- Dataset Description
- Bi-classification Task
- Top-k Task
- Explainability Analysis

Dataset Description

KKBox: A music recommendation dataset from Kaggle.

IM-1M: A movie recommendation dataset from IMDb and MovieLens-1M.

	Dataset	KKBox	IM-1M
User-Item Interaction	#Users	34,403	6,040
	#Items	2,296,320	3,274
	#Interactions	3,696,465	370,023
	Data Density	0.0047%	1.87%
Knowledge Graph	#Entities	2,562,937	15,439
	#Entities Types	5	5
	#Relation Types	8	9
	#Triples	16,237,068	442,409
Path	#Path	41,400,408	345,344
	Avg.Path.Length	5.11	4.74
	#Meta Path Types	21	46
	Avg.Meta.Path.Length	5	5.37

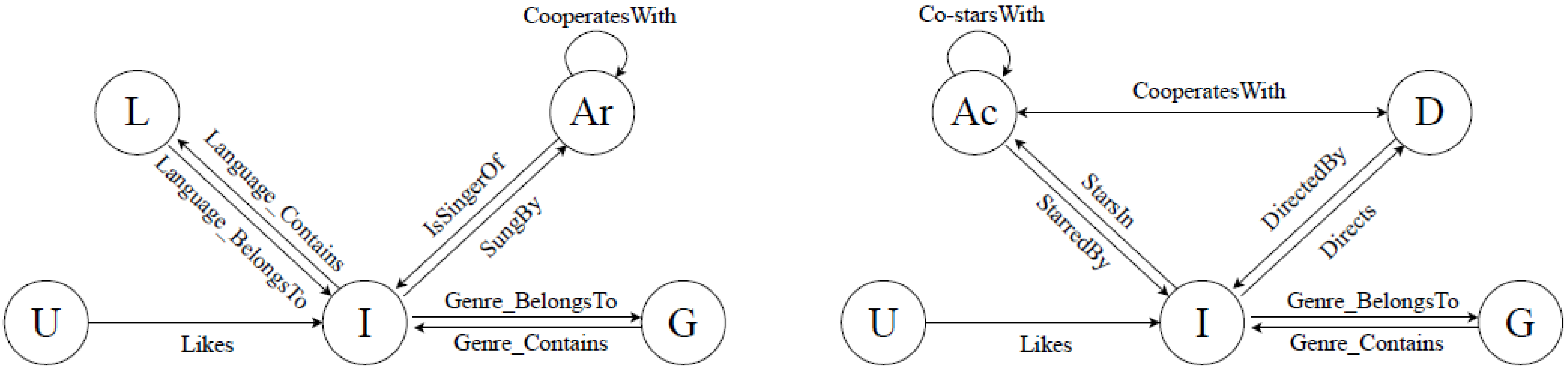


Figure 3: Schema graphs of KKBox (left) and IM-1M (right). In KKBox, U: user, I: item (song), L: language, Ar: artist, G: genre. In IM-1M, U: user, I: item (movie), Ac: actor, D: director, G: genre.

Both music and movie field knowledge graphs are manually constructed.

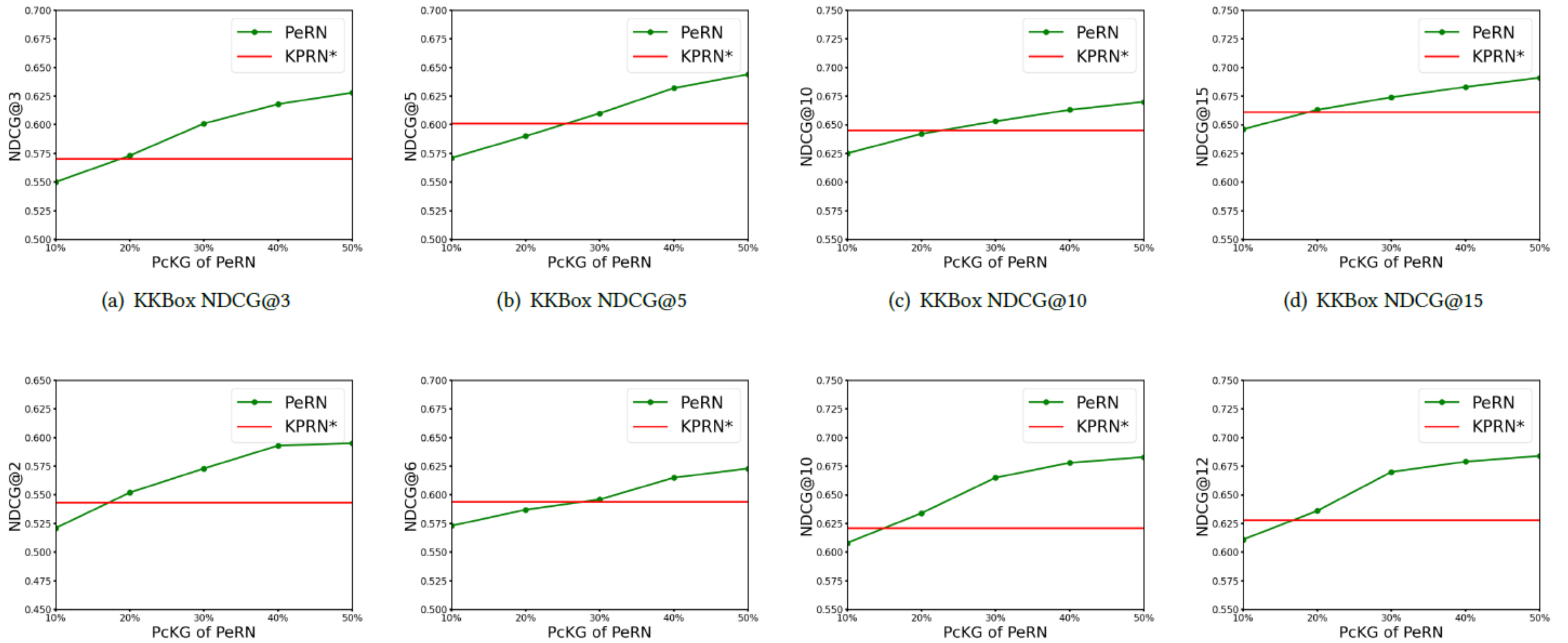
Here shows the schema graphs of KKBox and IM-1M knowledge graphs.

Bi-classification Task

Table 3: Summary of performance on binary classification recommendation task between all baselines and our proposed PeRN on KKBox and IM-1M datasetS. Bolded numbers indicate the best result of each columns, and '*' indicates the arithmetic square root operation is performed on MSE for simplicity.

Dataset	KKBox					IM-1M				
Metrics	P	R	F_1	MSE*	AUC	P	R	F_1	MSE*	AUC
MF	0.509	0.528	0.518	0.496	0.511	0.612	0.608	0.610	0.431	0.586
AFM	0.517	0.533	0.525	0.483	0.536	0.647	0.632	0.639	0.413	0.601
RippleNet	0.699	0.732	0.715	0.287	0.762	0.742	0.713	0.727	0.284	0.694
MEIRec	0.753	0.774	0.763	0.242	0.819	0.792	0.804	0.798	0.221	0.734
KPRN	0.805	0.822	0.813	0.206	0.834	**0.843**	0.826	0.834	0.172	0.812
PeRN	**0.842**	**0.861**	**0.851**	**0.195**	**0.866**	0.835	**0.871**	**0.853**	**0.154**	**0.851**

Top-k Task (on dealing cold-start issue)



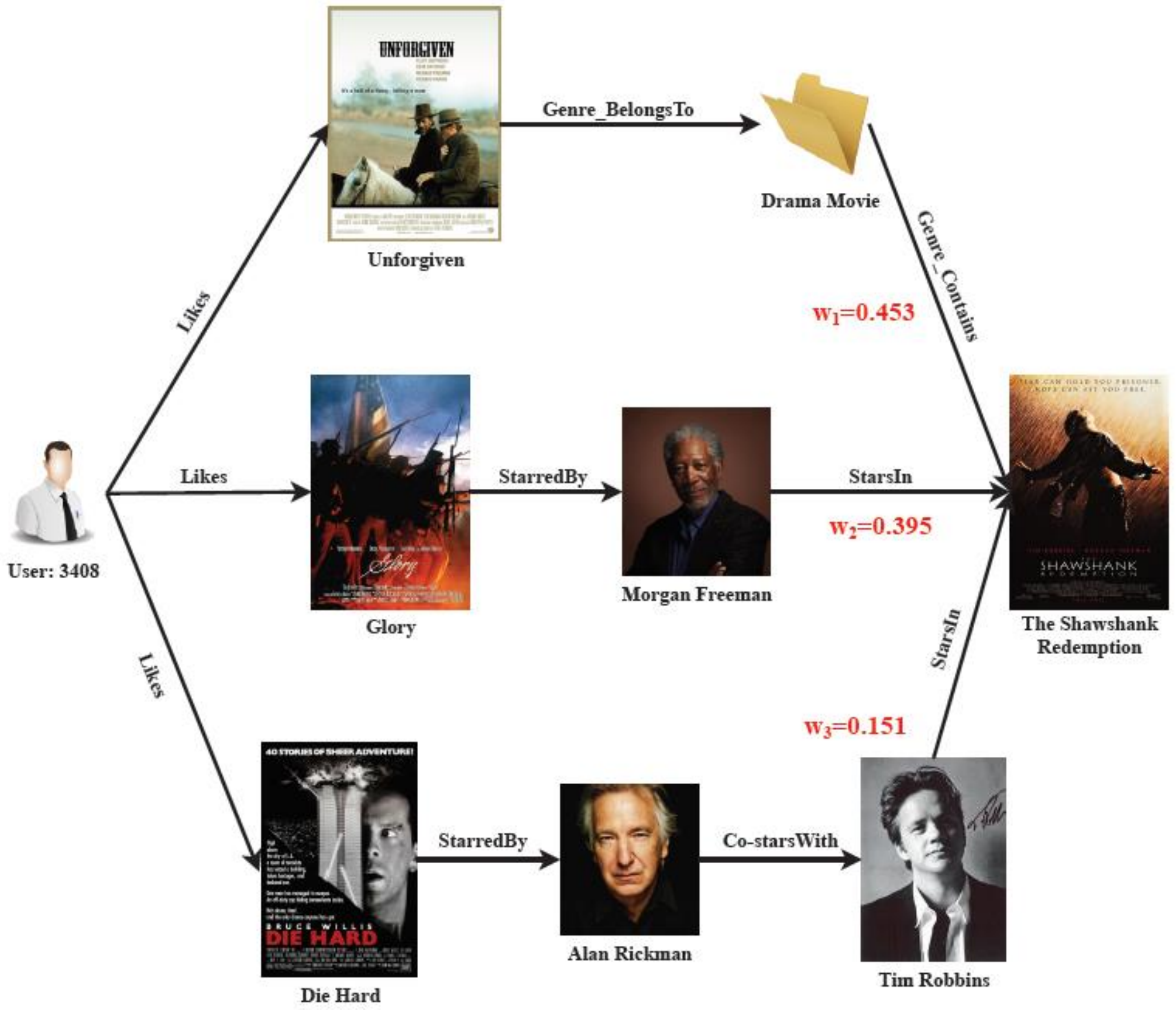
(a) KKBox NDCG@3 (b) KKBox NDCG@5 (c) KKBox NDCG@10 (d) KKBox NDCG@15

(e) IM-1M NDCG@2 (f) IM-1M NDCG@6 (g) IM-1M NDCG@10 (h) IM-1M NDCG@12

Figure 4: Performance of PeRN in top-K task and cold-start costs, measured by NDCG@{3, 5, 10, 15} in KKBox and NDCG@{2, 6, 10, 12} in IM-1M. '*' here indicates the PcKG of KPRN is constantly 50%.

PcKG: **P**ercentage of interactions used to **c**omplete **K**nowledge **G**raph

Explainability Analysis



References

- Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of UAI*. 452–461.
- Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional Factorization Machines: Learning the Weight of Feature Interactions via Attention Networks. In *Proceedings of IJCAI*. 3119–3125.
- Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems. In *Proceedings of CIKM*. 417–426.
- Shaohua Fan, Junxiong Zhu, Xiaotian Han, Chuan Shi, Linmei Hu, Biyu Ma, and Yongliang Li. 2019. Metapath-guided Heterogeneous Graph Neural Network for Intent Recommendation. In *Proceedings of SIGKDD*. 2478–2486.
- Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable Reasoning over Knowledge Graphs for Recommendation. In *Proceedings of AAAI*. 5329–5336.

Thanks for reading!

Slides made by Yafan Huang & Shihui Song