Reinforcement Learning over Sentiment-Augmented Knowledge Graphs towards Accurate and Explainable Recommendation

Sung-Jun Park*
Dong-Kyu Chae*
{tjdwns32,dongkyu}@hanyang.ac.kr
Hanyang University
Seoul, Republic of Korea

Hong-Kyun Bae
Hanyang University
Seoul, Republic of Korea
hongkyun@hanyang.ac.kr

Sumin Park
Hanyang University
Seoul, Republic of Korea
suminp@hanyang.ac.kr

Sang-Wook Kim†
Hanyang University
Seoul, Republic of Korea
wook@hanyang.ac.kr

ABSTRACT

Explainable recommendation has gained great attention in recent years. A lot of work in this research line has chosen to use the *knowledge graphs* (KG) where relations between entities can serve as explanations. However, existing studies have not considered *sentiment on relations* in KG, although there can be various types of sentiment on relations worth considering (e.g., a user's satisfaction on an item). In this paper, we propose a novel recommendation framework based on KG integrated with sentiment analysis for more accurate recommendation as well as more convincing explanations. To this end, we first construct a **S**entiment-**A**ware **K**nowledge **G**raph (namely, *SAKG*) by analyzing reviews and ratings on items given by users. Then, we perform item recommendation and reasoning over SAKG through our proposed **S**entiment-**A**ware **P**olicy **L**earning (namely, *SAPL*) based on a *reinforcement learning* strategy. To enhance the explainability for end-users, we further developed an *interactive user interface* presenting textual explanations as well as a collection of reviews related with the discovered sentiment. Experimental results on three real-world datasets verified clear improvements on both the accuracy of recommendation and the quality of explanations.

CCS CONCEPTS

• Information systems → Recommender systems; **• Computing methodologies → Knowledge representation and reasoning**.

KEYWORDS

Explainable recommendation, knowledge graph, sentiment analysis

*Both authors contributed equally to this research.
†Corresponding author.

WSDM '22, February 21–25, 2022, Tempe, AZ, USA.

ACM ISBN 978-1-4503-9132-0/22/02…$15.00
https://doi.org/10.1145/3488560.3498515

ACM Reference Format:
Sung-Jun Park, Dong-Kyu Chae, Hong-Kyun Bae, Sumin Park, and Sang-Wook Kim. 2022. Reinforcement Learning over Sentiment-Augmented Knowledge Graphs towards Accurate and Explainable Recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM '22), February 21–25, 2022, Tempe, AZ, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3488560.3498515

1 INTRODUCTION

The *recommender systems* (RS) have been extensively studied in the academic society as well as aggressively utilized by many companies such as Netflix and Amazon [9, 16]. Very recently, endowing RS the ability of offering *explanations* has gained great attention. Here, leveraging *knowledge graphs* (KG) is one of the preferred choices for developing such explainable RS. Figure 1 illustrates a toy example of KG and its reasoning. The KG consists of (*entity-relation-entity*) triplets where the entity can be a *user*, an *item* or a *feature*, and the relation can be a fact among the two entities like *purchase*, *described_by*, or *also_viewed*. A path from a user (e.g., *Julia*) to an item (e.g., *Galaxy S*) provides not only clues of recommendation but also intuitive stories of explanation; for instance, the path of $[Julia \xrightarrow{mention}$ "$anti\text{-}break$" $\xrightarrow{mentioned\ by} Tom \xrightarrow{purchase} Galaxy\ S]$ can be interpreted as "*Galaxy S* is recommended to *Julia* because she seems to care an item feature '*anti-break*', which is also mentioned by *Tom* who purchased *Galaxy S* ".

A brief history of this research area is as follows: early efforts tried to learn KG *embeddings* to get user and item representations and then to recommend items with the most similar embeddings to a target user [1, 14]. Then, post-hoc explanations are generated for the recommendations via a similarity-based path finding procedure. Recent researches are more focusing on exploring paths to retrieve items to be recommended over KG, where the path trajectory naturally delivers explanations. Notable examples include PGPR [34] and ADAC [37] that employed a *reinforcement learning agent* exploring paths on KG, and RippleNet [30], KPRN [33], and KGAT [32] that employed the *attention mechanism* to navigate entities and relations most contributing to the recommendation.

However, we argue that considering *sentiment* [12, 31, 36] in constructing KG can help in improving the accuracy of recommendation and the quality of explanations, but has been neglected by

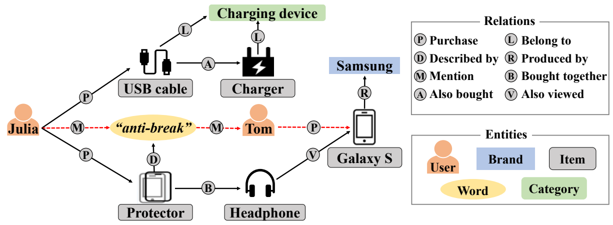


Figure 1: An illustration of a knowledge graph.

Figure 2: Benefits of considering users' sentiment.

existing work. Indeed, there can be various types of sentiment on relations between entities which are worth considering. For instance, (see Figure 1) the "*purchase*" relation typically defined between the user and item entities can be specified whether (1) "*satisfied*" that means a user preferred an item, (2) "*dissatisfied*", or (3) just remained "*purchase*" as neutral sentiment; the "*described_by*" relation defined between the item and feature entities can be specified whether (1) "*well_equipped*" that means an item has a strength in terms of a specific feature, (2) "*poorly_equipped*", or just "*described_by*" as neutral sentiment. Likewise, more fine-grained and sentiment-aware relations can be defined between various types of entities in KG and are expected not only to result in more accurate recommendation but also to make explanations more informative and trustworthy than just showing simple facts among entities.

To highlight our motivation, Figure 2 compares the recommendation results and path explanations provided by our work and existing work. First, **Case 1** shows that both existing work and our work recommended *Lace Boyshort Panty* to *Ben*, which was the correct answer, but our explanation path delivers more specific information via sentiment-aware relations such as "*mention_a_lot*" that means a user takes a specific feature (e.g., *lace*) importantly, and "*well_equipped*" that was explained before. Moreover, the actual review written by *Ben* to *Lace Boyshort Panty* confirmed that he liked the item mainly because of its nice "*lace*", which is an important feature that he cares about a lot. Next, **Case 2** shows that our work provided correct recommendation to *Max* owing to the sentiment information while the existing work reached to the wrong recommendation. The existing work selected *Julia* as an intermediate entity since she and *Max* had co-purchased *Chrome Hard Case*, but it seems a wrong choice since her review on it revealed *negative sentiment*. But our method selected *Tom*, which looks a good choice since his review shows *positive sentiment* on *Rubber Coated Case* that had also satisfied *Max*. Moreover, our explanation was more informative in this case.

Motivated by the clear benefits of considering sentiment on relations, we propose a novel explainable recommender system based on our proposed **S**entiment-**A**ware **K**nowledge **G**raph (*SAKG*, in short) and **S**entiment-**A**ware **P**olicy **L**earning (*SAPL*, in short). More specifically, this paper has the following main contributions:

(1) **SAKG**: We propose SAKG, a novel type of Knowledge Graph where various types of sentiment-related labels are assigned on relations among various types of entities. We analyze reviews and ratings of users on items to mine such sentiment signals (e.g., how much an item performs well from the viewpoint of a specific feature, and how much a user cares about a specific feature a lot). Such sentiment information also allows us to distinguish important words related with item features, frequently mentioned in users' reviews to express their sentiment. We employ those words as *word entities* of our SAKG, whereas the existing methods may misunderstand them as meaningless thus removing them due to their low TF-IDF scores.
(2) **SAPL**: To perform recommendation and reasoning over SAKG, we train a *reinforcement learning* agent navigating on our SAKG, where the sentiment information on the paths takes an important role of providing reward signals. Owing to their guidance, our agent can find a promising path to an item with high preference for a given user where the relations with positive sentiment are included as many as possible. For example, our agent starting from a user may select a relation labeled with "*mention_a_lot*" rather than choosing other options; then, if the word entity aligned with the chosen relation is linked to the item through a relation labeled with "*well_equipped*", the agent may reach the item. As a result, the agent successfully finds the item performing well in terms of the word (or, feature) that the user seems to care a lot. We also propose Sentiment-aware KG embedding to support the training of our agent.
(3) **Interactive user interface**: A path that our agent navigates from a target user to an item provides an intuitive explanation (its examples can be found in Figure 2). However, general end-users may still have difficulty in clearly understanding such path-based explanations. Some users may want to know about the underlying reason for the sentiment information included in the provided explanations. Our interactive user interface aims at making the explanations more user-friendly and informative. It not only provides textual explanations based on the recommendation path derived by our agent, but also shows a collection of critical reviews that originate the sentiment signals included in the explanations.

We evaluated our method on three types of Amazon e-commerce datasets. From our accuracy evaluation and ablation study, we confirmed that (1) our method achieves state-of-the-art accuracy and (2) each of proposed sentiment-oriented ideas contributes to enhancing the accuracy. We also performed two user studies, which demonstrated that (3) our method provides more-convincing explanations than other competitors and (4) our UI and its explanations are persuasive enough to gain users' trust and loyalty.

2 RELATED WORK

This section summarizes two research directions to develop explainable recommender systems (ERS, in short). We refer the reader to an excellent survey [35] for a review on ERS.

***ERS based on Knowledge Graph*:** A method proposed in [4] produces recommendations with their entity-level explanations by jointly ranking items and entities in KG by using a personalized PageRank. **KPRN** [33] learns representations of paths in KG by using the LSTM network to infer the underlying rationale of user-item interactions. A weighted pooling operation is also proposed to discriminate the strengths of different paths from a target user to an item. **PGPR** [34]] trains a reinforcement learning agent navigating on KG to find potential items and reasoning paths. **ADAC** [37] also leverages the reinforcement learning on KG where an agent finds demonstration-guided paths with minimum labeling efforts. However, they all overlooked various sentiment signals that may exist among entities in KG; we believe that considering sentiment between entities can improve both the recommendation accuracy and the explanation quality.

***ERS based on Attention*:** The attention scores assigned to components such as words/sentences in reviews or entities/relations in KG imply their relative contributions on the final recommendation. **D-Attn** [28] employs CNNs equipped with local and global attention layers to determine informative words in reviews that contribute to the positive rating. **CAML** [8] employs a hierarchical co-attentive selector to model the correlations between the recommendation task and the explanation task. **RippleNet** [30] propagates user preferences over the set of knowledge entities to predict clicking probability by iteratively extending a user's potential interests. **KGAT** [32] models the high-order relations in KG by recursively propagating embeddings of entities to their neighbors to update their embeddings, and then discriminates important neighbors of each entity with the attention mechanism. **KARN** [38] employs attention-based RNN to extract users' interest from their click sequences and make a reasoning over paths on KG between users and items. Despite the nice philosophy of the explanation with attention scores, we argue that the concept of attention would be familiar only to AI/ML developers but not to general end-users, meaning that customers would like to have scores with additional explanations indicating what those scores mean.

3 SENTIMENT-AWARE KNOWLEDGE GRAPH

This section elaborates how to construct the proposed SAKG. Formally, Let $\mathcal{G} = \left\{(e^h, r, e^t)|e^h, e^t \in \mathcal{E}, r \in \mathcal{R}\right\}$ be a SAKG where $\mathcal{E}$ indicates the set of entities, $\mathcal{R}$ indicates the set of relations among entities, and a triplet (e^h, r, e^t) indicates an observation that there is a relationship r (e.g., *produced_by*) from a head entity e^h (e.g., *iPhone3*) to a tail entity e^t (e.g., *Apple*). Similar with the existing work [1, 34, 37], we also exploit various resources of e-commerce data including (user, item, rating) triplets, reviews written by users to items, and meta-data of items, for defining entities and connecting them via relations. In addition, to incorporate users' sentiment into KG, we perform a sentiment analysis by using a method proposed from [20, 36]. It accepts the reviews as input and analyzes the grammar and sentence structure of review texts to output a sentiment lexicon $\mathcal{L} = \{(f, o, s)|f \in \mathcal{F},\ o \in \mathcal{O}, s \in \mathcal{S}\}$, where $\mathcal{F}$ denotes a set of words related with item features, $\mathcal{O}$ denotes a set of actual sentiment-related words (e.g., nice, good, and so on), and $\mathcal{S}$ does a set of labels {positive, negative} for the pair (f, o). For example, from a review text *"This skin fits perfect in my SGSII and the phone looks great inside of it"*, the triplets ('*fit*', '*perfect*', *positive*) and ('*look*', '*great*', *positive*) will be extracted.

Equipped with those resources, we now define five entity types for our SAKG: *user*, *item*, *brand*, *category*, and *word* in reviews [34]; so the corresponding data instances (e.g., users, items, brand names, category names, and words) constitute the entities of SAKG. Note that instead of using all the words in reviews as our *word* entities, we only consider the words (1) with a frequency less than 5,000 and with TF-IDF score > 0.1 by following [34], or (2) included in the set $\mathcal{F}$ (i.e., item features). The first condition helps in filtering frequent but less-meaningful words. However, we note that there can be some important words which are frequently mentioned in user reviews due to the domain characteristics of data. For example, in the Amazon *cellphone* dataset, users tend to reveal their sentiment on item features like *speaker*, *button*, *size*, *life*, or *volume*, which are apparently important words but will be eliminated by the first condition due to their high frequencies. Thanks to the second condition, however, they will not be filtered out, since the words are included in $\mathcal{F}$, becoming the entities of SAKG.

After defining entities, we now connect them with a directed relation if there is a relation from a head entity to a tail entity. Here, depending on types of the head and tail entities, we define various types of sentiment-aware relations as follows:

(1) *user→item*: From *user* entities to *item* entities, we define the following three types of sentiment-aware relations: *satisfied*, *dissatisfied*, and *purchase*. The ratings (in a scale of 1 to 5) are employed to identify the sentiment of users on items: the median score, 3, is considered as *purchase* (i.e., neutral sentiment), and the higher scores and lower scores are considered as *satisfied* and *dissatisfied*, respectively.

(2) *item→word*: From *item* entities to *word* entities, the sentiment-aware relations are defined based on how many positively/negatively a word has been mentioned: *well_equipped* for positive, *poorly_equipped* for negative, and *equipped* for neutral. For example, if many users have positively mentioned a cellphone's *speaker*, the relation *well_equipped* is defined from the *item* entity and the *word* ('*speaker*') entity. Note that each triplet (f, o, s) included in $\mathcal{L}$ obtained from our sentiment analysis contains such information: given an item i and a word w mentioned in i's reviews, we compare (1) the number of triplets where f equals to w and s is positive originated from i's reviews, and (2) the number of triplets where f equals to w and s is negative; if the number of positive/negative triplets is more, we define the positive/negative relation among the corresponding $i \to w$; if the two are the same number, we define neutral relation.

(3) *user→word*: If a user cares about a specific feature of items a lot, she may mention it frequently through reviews [36]. Following this intuition, the sentiment-aware relations from *user* entities to *word* entities are defined based on the word frequency: Among the words mentioned by a user, the top 10% based on their frequencies are considered as *mention_a_lot*; those mentioned only once are considered as *mention_barely*; the rest of words are considered as *mention* (i.e., neutral sentiment).

In our reinforcement learning framework which will be introduced in the next section, the relations labeled with *satisfied*, *well_ equipped*, and *mention_a_lot*[1] are considered as positive signals; *dissatisfied*, *poorly_ equipped*, and *mention_barely* are considered as negative signals; *purchase*, *equipped*, and *mention* are considered as neutral sentiment. We also allow many other types of relations including *also_bought*, *also_viewed*, and *bought_together* between *item* and *item* entities, *belonging_to* for *item* and *category* entities, and *produced_by* for *item* and *brand* entities [34]. However, we do not consider sentiment on those types of relations since they are just representing simple facts, but not revealing specific sentiment underlying them.

4 SENTIMENT-AWARE POLICY LEARNING

This section explains how to perform recommendation and reasoning over SAKG. For each user entity e_u in SAKG, our goal is to find a subset of item entities that u is likely to prefer by navigating the paths starting from e_u over SAKG. This task can be naturally formulated as a reinforcement learning problem where an agent is guided to find potential items for recommendation while taking into account the following relations with positive sentiment.

Before introducing our SAPL, we first perform embeddings of entities and relations in our SAKG, which will be further used for defining *states* and computing *rewards* at our reinforcement learning step. As proposed in [1], finding embeddings of entities and relations in KG is a task of maximizing the following likelihood:

$$\sum_{(e^h,r,e^t)\in\mathcal{G}} \log\sigma\left(\mathbf{e^t}\cdot(\mathbf{e^h}+\mathbf{r})\right) + k\cdot\mathbb{E}_{\tilde{e}^t\sim P_{type(e^t)}}\left[\log\sigma\left(-\tilde{\mathbf{e}}^{\mathbf{t}}\cdot(\mathbf{e^h}+\mathbf{r})\right)\right] \quad (1)$$

where the boldface notations $\mathbf{e^t}$, $\mathbf{r}$, $\mathbf{e^h}$ indicate embedding vectors of entities e^t, r, e^h, respectively. The first term attempts to locate e^t and e^h+r closely each other in the embedding space, while the second term does $\tilde{e}^t$ and e^h+r to be distant. Here, $\tilde{e}^t$ indicates a negative sample drawn from $P_{type(e^t)}$, which is noisy distribution of entities having the same type of e^t [23], and k indicates the size of negative sampling [5]. To consider sentiment among entities, we define a sentiment-aware weight w as follows:

$$w=\begin{cases}\alpha & ,\ if\ \mathcal{S}\left(e^h,r,e^t\right)=1\\ (2-\alpha), & if\ \mathcal{S}\left(e^h,r,e^t\right)=-1\\ 1 & ,\ if\ \mathcal{S}\left(e^h,r,e^t\right)=0\end{cases} \quad (2)$$

where $\mathcal{S}\left(e^h,r,e^t\right)$ indicates a simple function that returns 1, 0, or -1, according to the sentiment of the given triplet $\left(e^h,r,e^t\right)$: 1 for positive, -1 for negative, and 0 for neutral. α is a hyper-parameter between 0 and 1. Simply speaking, w will be less than 1 if a given relation r is positive, larger than 1 if negative, and 1 if neutral sentiment. This weight is multiplied to the embedding similarity $\mathbf{e^t}\cdot(\mathbf{e^h}+\mathbf{r})$ in the first term of the likelihood function as follows:

$$\sum_{(e^h,r,e^t)\in\mathcal{G}} \log\sigma\left(w\ \mathbf{e^t}\cdot(\mathbf{e^h}+\mathbf{r})\right) + k\cdot\mathbb{E}_{\tilde{e}^t\sim P_{type(e^t)}}\left[\log\sigma\left(-\tilde{\mathbf{e}}^{\mathbf{t}}\cdot(\mathbf{e^h}+\mathbf{r})\right)\right] \quad (3)$$

Since the value less than 1 is multiplied if the corresponding relation r is positive (e.g., *satisfied*), the similarity term $\mathbf{e^t}\cdot(\mathbf{e^h}+\mathbf{r})$ will be pushed to a higher value than before in order to maximize the entire objective function, which eventually makes $\mathbf{e^t}$ and $\mathbf{e^h}+\mathbf{r}$ close with each other. Similarly, the value larger than 1 is multiplied if the corresponding relation r is negative, which makes the similarity term $\mathbf{e^t}\cdot(\mathbf{e^h}+\mathbf{r})$ less pushed than before, thereby making $\mathbf{e^t}$ and $\mathbf{e^h}+\mathbf{r}$ less close to each other.

We now introduce our SAPL. Similar to the previous work [34, 37], we established MDP (Markov Decision Process) environment that an agent is interacting with over a sequence of discrete time steps $t=0,1,2,\ldots,T-1,T$. The following elaborates on some preliminaries of our MDP environment:

State. The state s at time t is expressed as $s_t=(\mathbf{u},\ \mathbf{e_t},\ \mathbf{h_t})$ where $\mathbf{u}$ indicates embedding of the starting user and $\mathbf{e_t}$ indicates the embedding of an entity where the agent is currently located. $\mathbf{h_t}$ indicates the history of all entities and relations visited prior to time t and expressed as a concatenation of their embedding vectors, $[\mathbf{e_{t-n}},\mathbf{r_{t-n+1}},\mathbf{e_{t-n+1}},\ldots,\mathbf{e_{t-1}},\mathbf{r_t}]$, where n indicates the length of a window size. Naturally, the initial state s_0 is expressed as $(\mathbf{u},\mathbf{u},)$. Note that the state vector at time t is also expressed as a concatenation of vectors inside: $\mathbf{s_t}=[\mathbf{u},\ \mathbf{e_t},\ \mathbf{h_t}]$.

Action. Given the current state $s_t=(\mathbf{u},\mathbf{e_t},\ \mathbf{h_t})$, our agent samples an action $a_t\in A_t$ where A_t indicates the action space and is defined as the set of all possible outgoing relations from e_t excluding those included already in history $\mathbf{h_t}$. Following the prior work [34, 37], we prune A_t with the maximum size 250 instead of maintaining the action space with the size based on the largest outdegree, which will be extremely inefficient. After taking an action $a_t=(r_{t+1},e_{t+1})$, the current state $s_t=(\mathbf{u},\mathbf{e_t},\ \mathbf{h_t})$ is transitioned to the next state $s_{t+1}=(\mathbf{u},\mathbf{e_{t+1}},\mathbf{h_{t+1}})$.

Reward. In our sentiment-aware setting, we encourage the agent to find promising paths for (1) providing convincing explanations to a target user and (2) reaching to potential item entities that she is highly likely to prefer. To achieve both objectives, the straightforward reward strategy would be to give a high reward, whenever the agent finds *positive* relations. However, this strategy may let the agent search for the positive relations only without trying to find out *item* entities for recommendation. Taking this risk into account, we define our reward at time t, R_t, as:

$$R_t=\ \beta R_t^{\mathcal{S}}+(1-\beta)\,R_t^{\mathcal{I}} \quad (4)$$

where the first sub-reward $R_t^{\mathcal{S}}$ focuses on whether the agent finds positive relations as many as possible at every time step, and the second sub-reward $R_t^{\mathcal{I}}$ is only given when the finally visited entity is an item. The tunable hyper-parameter β controls the weights on $R_t^{\mathcal{I}}$ and $R_t^{\mathcal{S}}$. $\mathcal{I}$ stands for a set of items. First, following [34], $R^{\mathcal{I}}$ is

[1] It means that a user "cares a lot" about a certain feature, rather than "likes" the feature. This is because people may like or dislike their purchased products in terms of a certain feature, but under both cases the feature will be mentioned frequently. One of our desired goals is to link a user with the words that she *cares a lot*, and then to link the words with the items performing well in terms of the feature.

computed as follows:

$$R_t^{\mathcal{I}} = \begin{cases} \max\left(0, \frac{\mathbf{e_t}\cdot(\mathbf{u}+\mathbf{r_p})}{max_{i\in\mathcal{I}}\ \mathbf{i}\cdot(\mathbf{u}+\mathbf{r_p})}\right), & if\ \ t=T\ \ and\ \ e_t\in\mathcal{I} \\ 0 & ,\ otherwise \end{cases} \tag{5}$$

As shown, the reward $R_t^{\mathcal{I}}$ is given only for the terminal state $s_T = (\mathbf{u}, \mathbf{e_T}, \mathbf{h_T})$ only if the agent reaches to one of item entities, thereby preventing our agent from always searching for positive relations only. The amount of reward is proportional to the similarity between the embedding vector of the item entity e_t visited at last ($t = T$) and the sum of embedding vectors of the given user u and the *purchase-based* relation r_p that means one of the three types of relations defined between users and items: '*satisfied*', '*dissatisfied*', and '*purchase*'.

Next, $R_t^{\mathcal{S}}$ is computed as:

$$R_t^{\mathcal{S}} = \begin{cases} \max\left(0, \frac{\mathbf{e_{t+1}}\cdot(\mathbf{e_t}+\mathbf{r_{t+1}})}{max_{e'\in\Omega(e_{t+1})}\ \mathbf{e'}\cdot(\mathbf{e_t}+\mathbf{r_{t+1}})}\right) & \\ & ,\ if\ \ \mathcal{S}(e_t, r_{r+1}, e_{t+1}) = 1 \\ 0 & ,\ otherwise \end{cases} \tag{6}$$

As shown, the reward $R_t^{\mathcal{S}}$ is given when the agent at $\mathbf{s_t} = (\mathbf{u}, \mathbf{e_t}, \mathbf{h_t})$ samples action $a_t = (r_{t+1}, e_{t+1})$ where the corresponding relation stands for positive (e.g., *satisfied* or *well_equipped*). $\Omega(e_{t+1})$ indicates the set of entities having the same type as e_{t+1}. This reward signal is proportional to the similarity between the embedding vector of the next entity e_{t+1} and the sum of embedding vectors of the current entity e_t and the chosen relation r_{t+1}.

Based on the aforementioned MDP environment, our agent learns its action policy denoted as $\pi(\cdot|\mathbf{s_t}, A_t)$, which outputs a probability distribution over actions to sample the next action, given the current state $\mathbf{s_t}$ and the set of all possible actions A_t. Basically, our policy aims to maximize the expected cumulative reward for any given user u:

$$Q(\theta) = \mathbb{E}_\pi\left[\sum_{t=0}^{T-1}\gamma^t\left(\beta R_{t+1}^{\mathcal{S}} + (1-\beta)\, R_{t+1}^{\mathcal{I}}\right)|s_0\right] \tag{7}$$

where γ indicates a discounting factor and θ indicates the set of model parameters for our policy network approximating π. Following [34], we employ REINFORCE with a baseline [29] algorithm to optimize θ, where the gradient for θ and Ψ, which is a set of model parameters for a value network employed in this algorithm, is computed by:

$$\nabla_{\theta,\Psi} Q(\theta,\Psi) = \mathbb{E}_\pi\left[\nabla_{\theta,\Psi}\log\pi_\theta(\cdot|\mathbf{s_t}, A_t)\,(G_t - \hat{v}_\Psi(\mathbf{s_t}))\right] \tag{8}$$

where G_t indicates the discounted cumulative reward at t, and $\hat{v}_\Psi(\mathbf{s_t})$ indicates the value network that outputs a real value used as the baseline in the REINFORCE algorithm, conditioned by $\mathbf{s_t}$.

After training the policy network, we perform the probabilistic beam search [34, 37] through our agent starting from a given user to find candidates for item recommendation. Among the candidate items, we pick a subset of items that make our agent receive the highest cumulative reward from the process of finding paths to them, and recommend them to the given user.

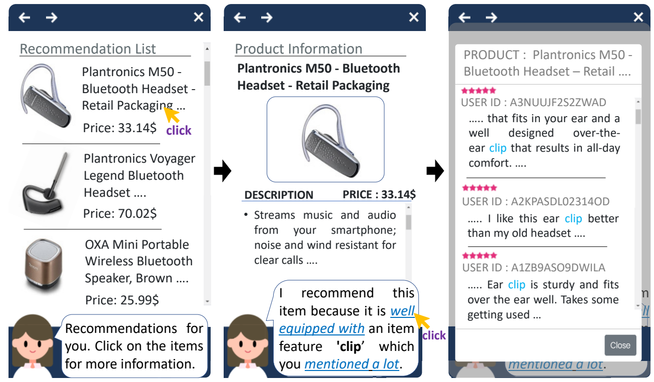


Figure 3: The snapshots of our UI. The text in this figure is enlarged more than it really is for readability.

5 INTERACTIVE USER INTERFACE

Even though a visualized path from a target user to an item suggested by the agent includes some information such as entity names and sentiment on relations that explain the reason for the recommendation, it still may not be easily understandable to naive end-users who may be more familiar with textual explanations. Hence, the users may have to put in a lot of efforts to interpret the path-based explanations, which may finally cause them to get tired of the system. In addition, some users would like to have more information about the explanations provided; for example, if a user is provided with an explanation with the sentiment signals "*well_equipped*" and "*mentioned_a_lot*", the user might wonder whether the suggested item performed well in terms of the feature (word) and if s/he really has mentioned the word a lot.

For the purpose of providing informative explanations which are more beneficial to naive users, we developed a prototype of an interactive user interface running on top of our reinforcement learning agent. It is implemented as a web-based, pop-up style agent displayed while a target user browses an online shop. Figure 3 shows several snapshots of our UI running on Amazon Cellphone data, one of our three real-world datasets used. Therefore, our agent in this example was learned based on our training set of the Amazon Cellphone data and the target user was chosen from our test set. A list of recommendations (see the leftmost screen) is provided to the user, and the user can click each item in the list. Corresponding explanations for each item are also presented, where the name of each entity and the label on each relation are highlighted (see the middle one). The textual explanation is generated based on our pre-defined templates depending on which types of entities and relations are included in the explanation path and how they are connected to each other.

In addition, users can click the sentiment information included in the textual explanation, which is highlighted with the text colored. Then, UI brings a pop-up window for showing a set of reviews that originate the corresponding sentiment (see the rightmost screen). For example, if the user clicks "*well equipped with*", s/he can take a glance at the list of reviews on the recommended item given by other (anonymized) users. Here, only the reviews that contributed to make the pair ('*clip*', *positive*) are shown. Also, if the user clicks "*mentioned a lot*", the user's reviews on other items including '*clip*'

Table 1: Dataset statistics.

Dataset	Cellphones	Clothings	CDs & Vinyls
#Users	27,879	39,387	75,258
#Items	10,429	23,033	64,443
#Interactions	194,439	278,677	1,097,592
#Entities in SAKG	163,249	425,528	581,099
#Tiplets in SAKG	4,149,296	6,287,078	29,333,872

are presented. This information, referring to the word ‘*clip*’ in reviews frequently, may help the user to realize the fact that s/he cares about the quality of ear clips on earphones a lot even if s/he has not been aware of this. In this way, our UI converts the agent’s reasoning over SAKG into more understandable and reliable explanations, which we believe will make end-users convince the recommender system more.

6 EVALUATION

6.1 Environment

6.1.1 Datasets. We used *Cellphones*, *Clothings*, and *CDs & Vinyl* datasets from the Amazon e-commerce data collection [22]. Table 1 shows their statistics. Each dataset contains user-item ratings and corresponding reviews, metadata of items such as categories and brands, and additional links among items such as *also_viewed* and *also_bought*, all of which were exploited to build our SAKG. We randomly split the interactions of each user into two subsets: the 70% for training and the remaining 30% for testing [34].

6.1.2 Competitors. We compared the recommendation accuracy of our framework with those of the following methods: **BPR** [27] is a matrix factorization model optimized by user-item pairwise ranking [19]; **BPR-HFT** (Hidden Factors and Topics) [21] is a review-based model that leverages topic distribution from text reviews to learn embeddings of users and items; **RippleNet** [30] propagates a user’s historical preferences on KG to enrich a user’s representation; **KGAT** [32] models users and items by considering their high-order connectivity on KG with the attention mechanism; **ADAC** [37] designs a *demonstration-based* KG reasoning framework by using reinforcement learning and imitation learning; **PGPR** [34] is also based on reinforcement learning to train an agent to find “good” paths for recommendation and reasoning over KG. Note that the last four methods (i.e., RippleNet, KGAT, ADAC and PGPR) can be categorized as ERS.

6.2 Recommendation Accuracy

We employed *NDCG* (Normalized Discounted Cumulative Gain), *Precision*, *Recall*, and *HR* (Hit Ratio) for evaluating each model’s **top-10** recommendation.

6.2.1 Comparisons with the state-of-the-arts. Table 2 reports the comparison results. Our method achieves the highest accuracy for *Cellphones* and *CDs & Vinyls* datasets, and shows a comparable accuracy for a *Clothings* dataset to that of ADAC. On average, our work improves 10.3% in accuracy over the best competitor in each case, and we believe that our idea of considering sentiment on relations led to such outperformance. BPR shows the lowest accuracy since it was trained only with user-item interactions without additional information (i.e., KG or reviews). BPR-HPT shows better performance than BPR since it additionally uses review texts. RippleNet shows higher performance than BPR-HPT owing to rich information in KG. Among PGPR, KGAT, and ADAC, ADAC shows the best performance on *Cellphones* and *Clothings* datasets, but its accuracy on *CDs & Vinyls* could not be obtained since its one time of training took more than 2 weeks. In addition, its optimal hyper-parameters for the *CDs & Vinyls* dataset are unknown, which made optimizing ADAC for this dataset even more difficult.

6.2.2 Ablation study. We performed ablation study by *removing* each of the three ideas from our method: (**W**) considering the sentiment-related words included in $\mathcal{F}$ when choosing words for defining *word* entities in our SAKG, (**E**) applying the sentiment-aware weight α to learn SAKG embedding, and (**R**) using the sentiment-aware reward signal $R_t^{\mathcal{S}}$ in SAPL. Table 3 shows the results. Only the results of Precision and HR on two datasets are reported here due to space limitations; the rest results indicated similar trend. We can observe that each sentiment-aware idea contributes to enhancing the performance of recommendation. In particular, the idea of keeping sentiment-related words in SAKG looks the most effective. We further inspected which words have been included as entities even though their frequencies are higher than 5,000, which are summarized in Table 4. Indeed, these words are essential to describe user satisfaction on items in their own domains, thereby mentioned in lots of reviews. Therefore, it seems reasonable to keep those words in SAKG rather than to regard them as meaningless words.

6.3 Explainability

Beyond the recommendation accuracy, we now aim to verify the superiority of our explanations in the human perspective. To this end, we performed two user studies having different objectives: one compares our explainability with that of the state-of-the-arts and the other comprehensively evaluates our explanation in terms of multiple aspects. Note that the detailed demographic information of the participants of each study can be found in Appendix. Several explanation cases are also provided in the last subsection.

6.3.1 Comparisons with the state-of-the-arts. In this study, we basically followed the protocols for user studies done by prior work in this research field [8, 18]: We randomly selected 30 comparative cases; each case includes three different explanation paths provided by methods *A* (SAPL), *B* (PGPR), and *C* (KGAT) (in random order, thus different for each case). We hired a set of 30 participants for the user study and asked the following question for each comparative case: “If you were the target user, which explanation seems more convincing to purchase the suggested item?”. Then, each participant can answer by *A*, *B*, *C* or *Give up (indistinguishable)*. Note that each comparative path looks similar in the upper case of Figure 2: their paths start from the same target user and reach the same item (recommendation); the only difference is the configuration of the recommendation paths. The paths provided by KGAT include the computed attention scores together with the labels on relations.

We’d like to emphasize that, in this user study, we did not use our UI to present our explanations; instead, we just showed the visualized explanation paths as the other competitors who do not

Table 2: Comparison results.

Datasets	Cellphones				Clothings				CDs & Vinyls			
Metrics	NDCG	Precision	Recall	HR	NDCG	Precision	Recall	HR	NDCG	Precision	Recall	HR
BPR	1.998	0.595	3.258	5.273	0.601	0.185	1.046	1.767	2.009	1.085	2.679	8.554
BPR-HPT	3.151	0.860	5.307	8.125	1.067	0.297	1.819	2.872	2.661	1.268	3.570	9.926
Ripple Net	4.837	1.101	7.716	11.454	2.195	0.603	3.892	6.032	4.871	1.852	7.145	15.727
ADAC	5.220	1.358	8.943	12.537	**3.048**	**0.783**	5.152	7.502	-	-	-	-
KGAT	5.111	1.296	8.978	12.589	3.021	0.747	5.172	7.394	5.411	2.120	7.764	17.173
PGPR	5.042	1.274	8.416	11.904	2.858	0.728	4.834	7.020	5.590	2.157	7.545	16.774
Ours	**5.710**	**1.443**	**9.589**	**13.440**	3.043	0.782	**5.190**	**7.541**	**6.941**	**2.745**	**9.551**	**20.524**
Improvement (%)	**9.4**	**6.3**	**6.8**	**6.8**	-0.2	-0.1	**0.4**	**0.5**	**24.2**	**27.3**	**23.0**	**19.5**

Table 3: Results for ablations.

Dataset	Cellphones		Clothings	
Metrics	Precision	HR	Precision	HR
Ours w/o W & E & R	1.274	11.904	0.728	7.02
Ours w/o E & R	1.404	13.107	0.775	7.459
Ours w/o R	1.422	13.343	0.778	7.523
Ours	**1.443**	**13.44**	**0.782**	**7.541**

Table 4: Sentiment-related words, ranked by frequency.

Dataset	Cellphones	Clothings	CDs
1	phone	fit	album
2	case	size	song
3	use	shoe	music
4	charge	wear	cd
5	battery	look	sound
6	screen	price	track
7	iphone	pair	band

have their own UI. By doing so, we made the comparison as fair as possible and focused on the pure effectiveness of our idea of incorporating sentiment into KG.

Figure 4 reports the comparative results, where the left side shows the percentage of cases chosen from the majority of participants among the 30 cases and the right side shows the percentage of choices from all participants for all cases. We observed that our method outperformed both PGPR and KGAT; (1) in 19 cases (63.3%) the majority of the participants chose our explanation to be best and (2) among 900 (30×30) choices, our method was chosen as the best one 394 times (43.8%). We conjecture this result is because the sentiment signals provided by ours are more informative and persuasive than the attention scores provided by KGAT and the relation types provided by PGPR without sentiment. We also observed that KGAT outperformed PGPR; it might be due to the difference that KGAT shows both attention scores and relation types while PGPR only shows relation types.

6.3.2 In-depth analysis of our explanations with UI. Next, we conducted one more user study to comprehensively evaluate our explanations aided by our UI. We hired another set of 30 participants disjointly from those in the previous user study. Since the UI is implemented on top of our three Amazon datasets, we randomly assigned several users in the test set of each data to the participants. Then, the participants browsed each recommended item and the corresponding explanations from the viewpoint of each assigned user. Afterwards, they were asked to evaluate the presented explanations and the UI with our questionnaire including the nine questions as below:

(1) *I understood easily why this system recommended this item to the target user.*
(2) *I thought the provided explanation made the recommendation more convincing.*
(3) *I thought the provided explanation made the target user trust the system more.*
(4) *I found the system provided useful information.*
(5) *I thought that I would like to use this system.*
(6) *I thought the system was easy to use.*
(7) *I found the system was unnecessarily complex.*
(8) *I thought there was too much inconsistency in this system.*
(9) *I thought this system was designed well.*

We composed the above questions based on the questionnaire of ResQue [3, 7, 15, 24, 26] and System Usability Scale (SUS) [2, 10, 13], which are widely-used evaluation frameworks for recommender systems and interfaces, mainly in the field of HCI [11, 17, 25]. The first four questions cover the important aspects of ERS: (1) *transparency*, (2) *persuasiveness*, (3) *trustworthiness*, and (4) *informativeness*. The next four questions are related with the usability of UI, including (5) *re-usability*, (6) *ease of use*, (7) *layout adequacy*, and (8) *consistency*. The last question (9) asks overall satisfaction. All the questions were in the form of a five-point scale and the answers ranged from 1 (strongly disagree) to 5 (strongly agree). Note that the lower the better for (7) layout adequacy and (8) consistency, while the higher the better for the others.

Figure 5 reports the result on each question, where the length of each bar represents the average of the scores given by the 30 participants, and the black line represents the 95% confidence interval. We observed that all questions were evaluated positively. In particular, the questions evaluating our explainability such as persuasiveness (avg=4.53, std=0.63), informativeness (avg=4.37, std=0.56), and transparency (avg=4.43, std=0.63) received very high scores which are greater than 4.0. This result again confirmed the superiority of our explanations, which are upgraded by our UI. As a result, our UI received a good score (avg=4.1, std=0.61) in terms of the overall

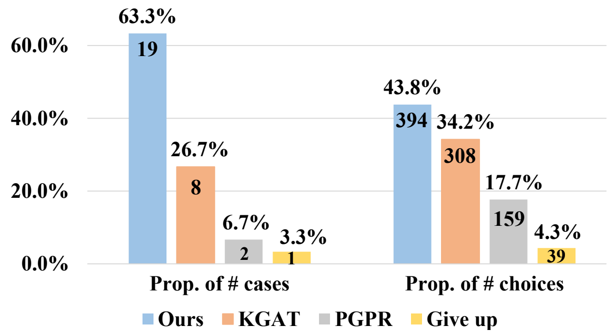


Figure 4: Results for our user study on explainability.

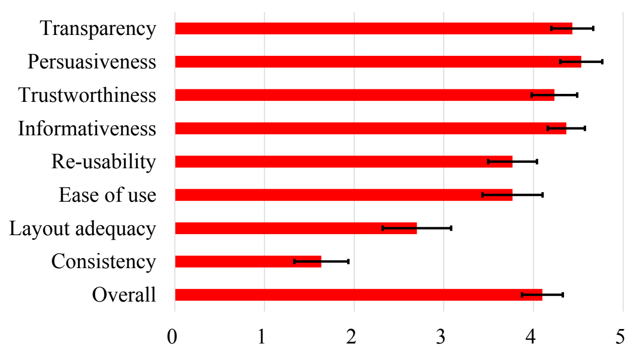


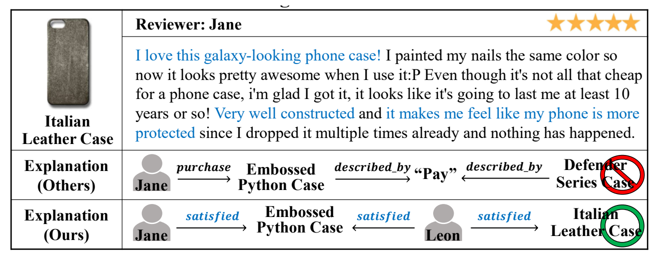
Figure 5: Results for our user study on UI.

satisfaction, and we believe the way how our UI explains the reason for recommendation mainly contributed to this positive result.

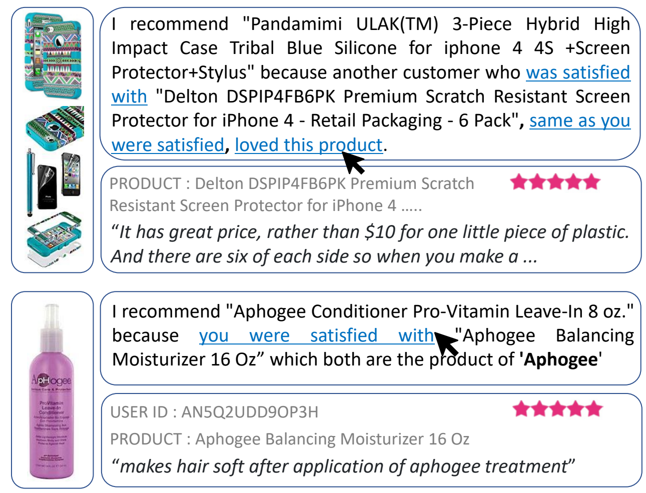
6.3.3 Case study. We believe Figure 2 shown in the first section has sufficiently appealed the superiority of our explanations equipped with sentiment information. Note that the two explanation paths compared in Figure 2 were the *actual outputs* from ours and PGPR obtained when they were recommended to the same target user. Figure 6(a) shows another case for comparison. Again, we note our work provided correct recommendation to *Jame* owing to the sentiment information while PGPR reached the wrong recommendation. Both ours and PGPR selected '*Embossed Python Case*' as the first hop. However, our agent then selected *Leon* who was satisfied with both '*Embossed Python Case*' and the recommended item, while PGPR selected a word entity '*Pay*' which was just aligned with both '*Embossed Python Case*' and the recommended item. As a result, our method successfully provided accurate recommendation as well as trustworthy explanations by considering sentiment. Figure 6(b) shows more examples of the explanations provided by our UI. The textual explanation in the upper case is based on a recommendation path including another customer who had similar satisfaction with the target user on both the recommended item and another item. The second explanation is based on a path including a maker and its product that the target user seems to prefer. In both cases, our UI provides the related reviews if the target user clicks the corresponding sentiment signals.

7 CONCLUSIONS

This paper explored the use of sentiment on the task of explainable recommendation with KG. First, a new type of KG, named SAKG, was proposed to augment sentiment information. To capture users' sentiment on various types of entities, we exploited the user-item ratings and the results from the sentiment analysis on reviews. Then, we proposed a novel reinforcement learning framework named SAPL, which guides our agent via carefully-designed reward signals to achieve two objectives: finding users' preferred items and choosing positive relations in SAKG. To support the training of our agent, we also proposed sentiment-aware embedding for SAKG. We further developed an interactive user interface that helps users easily understand the provided explanations as well as capture underlying reasons for the sentiment signal included in explanations. We performed extensive experiments on three real-world datasets (i.e., comparative and ablation studies) for evaluating the accuracy of recommendation, and two user studies for evaluating the quality of explainability. The experimental results from both perspectives demonstrated the superiority of our method over the state-of-the-arts.



(a) Explainability comparisons



(b) More explanations provided by our UI

Figure 6: Case study.

ACKNOWLEDGMENTS

This work was partly supported by Institute of Information communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2020-0-01373,Artificial Intelligence Graduate School Program(Hanyang University)) and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2020R1A2B5B03001960 and No. 2021R1A2C1094863).

REFERENCES

[1] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. 2018. Learning Heterogeneous Knowledge base Embeddings for Explainable Recommendation. *Algorithms* 11, 9 (2018), 137.

[2] John Brooke et al. 1996. SUS-A Quick and Dirty Usability Scale. *Usability Evaluation in Industry* 189, 194 (1996), 4–7.

[3] Bruno De Lemos Ribeiro Pinto Cardoso, Gayane Sedrakyan, Francisco Gutiérrez, Denis Parra, Peter Brusilovsky, and Katrien Verbert. 2019. IntersectionExplorer, a Multi-Perspective Approach for Exploring Recommendations. *International Journal of Human-Computer Studies* 121 (2019), 73–92.

[4] Rose Catherine, Kathryn Mazaitis, Maxine Eskénazi, and William W. Cohen. 2017. Explainable Entity-based Recommendations with Knowledge Graphs. In *Workshop Proceedings of the 11th Conference on Recommender Systems (RecSys)*.

[5] Dong-Kyu Chae, Jin-Soo Kang, Sang-Wook Kim, and Jung-Tae Lee. 2018. CFGAN: A Generic Collaborative Filtering Framework based on Generative Adversarial Networks. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM)*. ACM, 137–146.

[6] Dong-Kyu Chae, Sang-Wook Kim, and Jung-Tae Lee. 2019. Autoencoder-based Personalized Ranking Framework Unifying Explicit and Implicit Feedback for Accurate Top-N Recommendation. *Knowledge-Based Systems* 176 (2019), 110–121.

[7] Li Chen and Feng Wang. 2017. Explaining Recommendations Based on Feature Sentiments in Product Reviews. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces (IUI)*. ACM, 17–28.

[8] Zhongxia Chen, Xiting Wang, Xing Xie, Tong Wu, Guoqing Bu, Yining Wang, and Enhong Chen. 2019. Co-Attentive Multi-Task Learning for Explainable Recommendation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*. 2137–2143.

[9] Kyung-Jae Cho, Yeon-Chang Lee, Kyungsik Han, Jaeho Choi, and Sang-Wook Kim. 2019. No, That's Not My Feedback: TV Show Recommendation Using Watchable Interval. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 316–327.

[10] Yaakov Danone, Tsvi Kuflik, and Osnat Mokryn. 2018. Visualizing Reviews Summaries as a Tool for Restaurants Recommendation. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces (IUI)*. ACM, 607–616.

[11] Guglielmo Faggioli, Mirko Polato, and Fabio Aiolli. 2019. Tag-Based User Profiling: A Game Theoretic Approach. In *Proceedings of the 27th Conference on User Modeling, Adaptation and Personalization (UMAP)*. ACM, 267–271.

[12] Jingyue Gao, Xiting Wang, Yasha Wang, and Xing Xie. 2019. Explainable Recommendation through Attentive Multi-View Learning. In *Proceedings of the 33rd Conference on Artificial Intelligence (AAAI)*. AAAI Press, 3622–3629.

[13] Marko Gasparic, Andrea Janes, Francesco Ricci, and Marco Zanellati. 2017. GUI Design for IDE Command Recommendations. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces (IUI)*. ACM, 595–599.

[14] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y Chang. 2018. Improving Sequential Recommendation with Knowledge-Enhanced Memory Networks. In *Proceedings of the 41st International Conference on Research & Development in Information Retrieval (SIGIR)*. ACM, 505–514.

[15] Bart P. Knijnenburg, Martijn C. Willemsen, Zeno Gantner, Hakan Soncu, and Chris Newell. 2012. Explaining the User Experience of Recommender Systems. *User Modeling and User-Adapted Interaction* 22, 4-5 (2012), 441–504.

[16] Yunyong Ko, Jae-Seo Yu, Hong-Kyun Bae, Yongjun Park, Dongwon Lee, and Sang-Wook Kim. 2021. MASCOT: A Quantization Framework for Efficient Matrix Factorization in Recommender Systems. In *Proceedings of the 21st IEEE International Conference on Data Mining (ICDM)*. IEEE, 290–299.

[17] Johannes Kunkel, Benedikt Loepp, and Jürgen Ziegler. 2017. A 3D Item Space Visualization for Presenting and Manipulating User Preferences in Collaborative Filtering. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces (IUI)*. ACM, 3–15.

[18] Trung-Hoang Le and Hady W. Lauw. 2020. Synthesizing Aspect-Driven Recommendation Explanations from Reviews. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*. 2427–2434.

[19] Yeon-Chang Lee, Taeho Kim, Jaeho Choi, Xiangnan He, and Sang-Wook Kim. 2021. M-BPR: A Novel Approach to Improving BPR for Recommendation with Multi-Type Pair-Wise Preferences. *Information Sciences* 547 (2021), 255–270.

[20] Yue Lu, Malú Castellanos, Umeshwar Dayal, and ChengXiang Zhai. 2011. Automatic Construction of a Context-Aware Sentiment Lexicon: An Optimization Approach. In *Proceedings of the 20th International Conference on World Wide Web (WWW)*. ACM, 347–356.

[21] Julian McAuley and Jure Leskovec. 2013. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. In *Proceedings of the 7th Conference on Recommender Systems (RecSys)*. ACM, 165–172.

[22] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based Recommendations on Styles and Substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 43–52.

[23] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 27th Conference on Artificial Intelligence (AAAI)*. 3111–3119.

[24] Martijn Millecamp, Nyi Nyi Htun, Cristina Conati, and Katrien Verbert. 2019. To Explain or Not to Explain: The Effects of Personal Characteristics When Explaining Music Recommendations. In *Proceedings of the 24th International Conference on Intelligent User Interfaces (IUI)*. ACM, 397–407.

[25] Denis Parra, Peter Brusilovsky, and Christoph Trattner. 2014. See What You Want to See: Visual User-Driven Approach for Hybrid Recommendation. In *Proceedings of the 19th International Conference on Intelligent User Interfaces (IUI)*. ACM, 235–240.

[26] Pearl Pu, Li Chen, and Rong Hu. 2011. A User-Centric Evaluation Framework for Recommender Systems. In *Proceedings of the Fifth ACM Conference on Recommender systems (RecSys)*. 157–164.

[27] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (AAAI)*. 452–461.

[28] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable Convolutional Neural Networks with Dual Local and Global Attention for Review Rating Prediction. In *Proceedings of the 11th Conference on Recommender Systems (RecSys)*. ACM, 297–305.

[29] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement Learning - An Introduction.* MIT press, Cambridge, MA, USA.

[30] 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 the 27th International Conference on Information and Knowledge Management (CIKM)*. ACM, 417–426.

[31] Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. 2018. Explainable Recommendation via Multi-Task Learning in Opinionated Text Data. In *Proceedings of the 41st International Conference on Research & Development in Information Retrieval (SIGIR)*. ACM, 165–174.

[32] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge Graph Attention Network for Recommendation. In *Proceedings of the 25th International Conference on Knowledge Discovery & Data Mining (SIGKDD)*. ACM, 950–958.

[33] 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 the 33rd Conference on Artificial Intelligence (AAAI)*. 5329–5336.

[34] Yikun Xian, Zuohui Fu, S Muthukrishnan, Gerard De Melo, and Yongfeng Zhang. 2019. Reinforcement Knowledge Graph Reasoning for Explainable Recommendation. In *Proceedings of the 42nd International Conference on Research & Development in Information Retrieval (SIGIR)*. ACM, 285–294.

[35] Yongfeng Zhang and Xu Chen. 2020. Explainable Recommendation: A Survey and New Perspectives. *Foundations and Trends in Information Retrieval* 14, 1 (2020), 1–101.

[36] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit Factor Models for Explainable Recommendation based on Phrase-Level Sentiment Analysis. In *Proceedings of the 37th International Conference on Research & Development in Information Retrieval (SIGIR)*. ACM, 83–92.

[37] Kangzhi Zhao, Xiting Wang, Yuren Zhang, Li Zhao, Zheng Liu, Chunxiao Xing, and Xing Xie. 2020. Leveraging Demonstrations for Reinforcement Recommendation Reasoning over Knowledge Graphs. In *Proceedings of the 43rd International Conference on Research & Development in Information Retrieval (SIGIR)*. ACM, 239–248.

[38] Qiannan Zhu, Xiaofei Zhou, Jia Wu, Jianlong Tan, and Li Guo. 2020. A Knowledge-Aware Attentional Reasoning Network for Recommendation. In *Proceedings of the 34th Conference on Artificial Intelligence (AAAI)*. AAAI Press, 6999–7006.

A APPENDIX

A.1 Implementation Details

Since PGPR and ADAC are based on reinforcement learning as our work, we mainly refer to their parameter settings. When learning embedding of our SAKG, we used the batch size of 64, the embedding size of 100, the learning rate of 0.5, and the number of epochs of 30. The sentiment-aware weight parameter α was set to 0.9, 0.6, and 0.5 for *Clothings*, *Cellphones*, and *CDs & Vinyls*, respectively. For the MDP environment, the window size for the history vector $\mathbf{h_t}$ was set to 1. β was set to 0.02 for *Clothings* and *CDs & Vinyls* and 0.1 for *Cellphones*. The discount factor γ was set to 0.99. For SAPL, the number of epochs was set to 50, the learning rate was set to 0.0005 for *Cellphones* and *Clothings* and 0.001 for *CDs & Vinyls*, the batch size was set to 16 for *Cellphones* and 64 for *Clothings* and *CDs & Vinyls*. The maximum length for the recommendation path was set to 3. Following PGPR, we also allowed reverse edges and self loops in our SAKG when our agent explores recommendation paths. Path diversification via the action dropout [34, 37] was also employed with a rate of 0.5. We also borrowed several ideas employed in PGPR and ADAC: we also allowed reverse edges and self loops in our SAKG when our agent explores recommendation paths.

A.2 Evaluation Metric Details

We employed *NDCG* (Normalized Discounted Cumulative Gain), *Precision*, *Recall*, and *HR* (Hit Ratio).

First, for a user u, we employ NDCG to reflect the importance of ranked positions of items in N_u (i.e., a set of N recommended items for u). Let y_k represent a binary variable for the k-th item i_k in N_u (i.e., $y_k \in \{0,1\}$). If $i_k \in Rel_u$, y_k is set as 1, where Rel_u denotes a set of items considered relevant (i.e., ground truth). Otherwise, y_k is set as 0. In this case, $NDCG_u@N$ is computed by:

$$NDCG_u@N = \frac{DCG_u@N}{IDCG_u@N}$$

$$DCG_u@N = \sum_{k=1}^{N} \frac{2^{y_k} - 1}{\log_2^{(k+1)}}$$

where $IDCG_u@N$ is the ideal DCG at N (i.e., for every item i_k in N_u), y_k is set as 1.

Next, $Precision_u@N$ and $Recall_u@N$ are computed as follows:

$$Precision_u@N = \frac{|Rel_u \bigcap N_u|}{|N_u|}$$

$$Recall_u@N = \frac{|Rel_u \bigcap N_u|}{|Rel_u|}$$

Finally, $HR_u@N$ is defined by the following equation:

$$HR_u@N = \frac{|\{u||Rel_u \bigcap N_u| > 0\}|}{|\mathcal{U}|}$$

where $\{u||Rel_u \bigcap N_u| > 0\}$ indicates the set of users who receives the correct results and $\mathcal{U}$ denotes the set of all users [6].

As mentioned in the main manuscript, we set N as 10. After recommending top-10 items to the users, we compute those $NDCG_u@10$, $Precision_u@10$, $Recall_u@10$ and $HR_u@10$ and then average them. Each averaged score is denoted as *NDCG*, *Precision*, *Recall* and *HR* in Tables 2 and 3 of the main manuscript.

A.3 Statistics for SAKG

Tables A.1 and A.2 show detailed statistics for SAKG, where Table A.1 summarizes statistics for the entities and Table A.2 does statistics for the relations associated with SAKG for each dataset.

Table A.1: Statistics for SAKG entities

Dataset	Cellphones	Clothings	CDs & Vinyls
User	27,879	39,387	75,258
Feature	22,493	21,366	202,959
Product	10,429	23,033	64,443
Category	206	1,193	770
Brand	955	1,182	1,414
Related_product	101,287	339,367	236,255
Total entities	**163,249**	**425,528**	**581,099**

Table A.2: Statistics for SAKG relations

Dataset	Cellphones	Clothings	CDs & Vinyls
Satisfied	114,796	170,773	662,034
Neutral	16,462	23,339	73,978
Dissatisfied	18,790	20,584	68,078
Total Purchase	**150,048**	**214,696**	**804,090**
Mention_a_lot	98,970	73,731	1,415,233
Neutral	201,989	207,733	1,487,340
Mention_barely	1,667,337	2,090,662	10,008,080
Total Mention	**1,968,296**	**2,372,126**	**12,910,653**
Well_equipped	355,230	430,810	5,633,706
Neutral	985,355	1,514,916	4,925,821
Poorly_equipped	37,678	21,960	826,487
Total Described by	**1,378,263**	**1,967,686**	**11,386,014**
Belong_to	36,393	154,833	466,951
Produced_by	5,418	3,964	13,381
Also_bought	589,504	1,413,142	3,691,596
Also_viewed	12,930	144,794	17,424
Bought_together	8,444	15,837	44,076
Total relations	**4,149,296**	**6,287,078**	**29,334,185**

A.4 Participant Details

Table A.3 shows the detailed demographic information of the participants of each user study.

Table A.3: Demographic information of the participants.

		User study 1	User study 2
Gender	Male	16 (53%)	18 (60%)
	Female	14 (47%)	12 (40%)
Age	22-25	13 (43%)	24 (80%)
	26-29	14 (47%)	2 (7%)
	30-36	3 (10%)	4 (13%)
	Total	30	30