# Mutual Learning between Reviews and Ratings for Recommendation with Knowledge-based Neural Network

Yun LIU[†] and Jun MIYAZAKI[†]

[†] School of Computing, Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo 152-8550 Japan
E-mail: †liu.y.bp@m.titech.ac.jp, ††miyazaki@cs.titech.ac.jp

**Abstract** In many current recommender systems, users' reviews are used to boost the recommendation performance. As historical ratings and reviews are the two main feedbacks of users, the combination of these two behaviors is very important to understand why a user likes or dislikes an item. A general way of learning the correlation between reviews and ratings is to learn the distributions of them in the same latent semantic space, where each dimension is a set of features of items. However, there are only a few words related to items in reviews, the method may introduce noise and give wrong understanding. Besides, it is difficult to align heterogeneous data using the general way. In this paper, we introduce a knowledge graph to help focus on those reviews' entities that are relevant with target items, and jointly model reviews and ratings from shared knowledge spaces.

**Key words** Review-based Recommender Systems, Mutual Learning, Deep Learning, Knowledge graph

## 1 Introduction

Recently, the mutual learning between ratings and reviews for recommendation is researched by many works [5], [12], [4]. These works mainly focus on topic extracting from reviews [4], or using word embeddings to represent reviews [2]. They usually fuse ratings and reviews by defining transaction functions or using nonlinear neural networks. However, ratings and reviews are heterogeneous data. It is difficult to align their latent features using current methods. Thus, we fuse ratings and reviews in the same knowledge graph to solve the above problem. Besides, a knowledge graph can extract entities which related with items, and also can explore user potential preferences.

In this paper, we propose a framework named **R**ating Review **M**utual Learning with **K**nowledge Graph (RMK) for the high-quality recommendation. There are three components in RMK, which are fusion knowledge graph construction, knowledge graph embedding model, and rating prediction model. The main contribution of this paper as follows:

- To the best of our knowledge, we are the first work to fuse ratings and reviews through map review entities and user-item ratings to the same knowledge graph.
- The proposed model RMK is an end-to-end framework and can learn two different tasks simultaneously to enhance the correlations between ratings and reviews.

## 2 Related Works

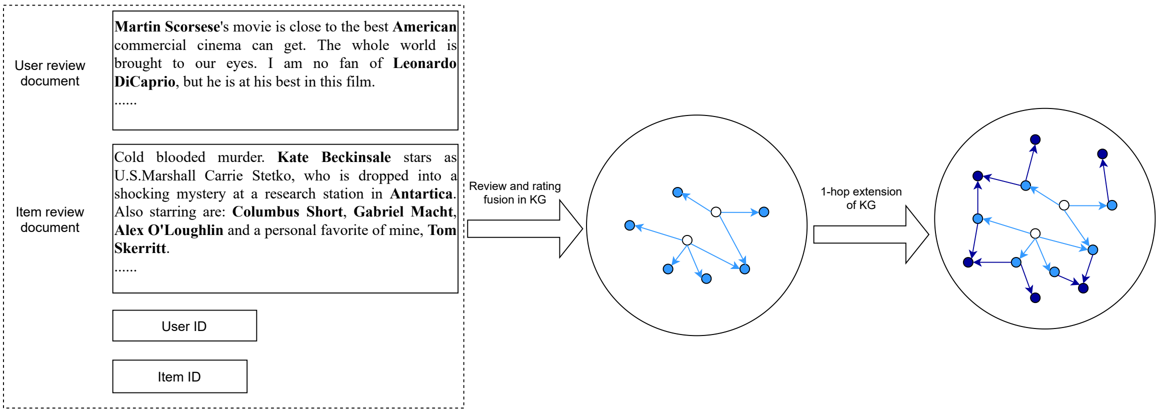
Some works have attempted to study the mutual learning between ratings and reviews for alleviating the cold-start problem in traditional recommender systems. They can be summarized into three types, topic-based methods, deep neural networks, and knowledge-based models.

### 2.1 Topic-based Works

The topic-based models are extensively used for dealing with user reviews in recommender systems. Through extracting latent topics or sentiments from reviews and aligning with the latent dimensions in ratings, the models could learn user preferences and item properties [1], [4], [7], [13], [14], [15]. In particular, McAuley et al. [4] used Latent Dirichlet Allocation (LDA) to learn latent topics from reviews and defined a transformation function to link hidden factors and hidden topics (HFT) together. Ling et al. [13] adopted topic models on reviews and used a mixture of Gaussian to learn latent features from ratings (RMR). Tan et al. [7] proposed a rating-boosted framework (RBLT) to enhance the topic features with higher ratings. They used the topic distributions of the item to model the preference distribution of users who rated the item. However, reviews and ratings are heterogeneous data, which is difficult to align them through the linear transformation.

### 2.2 Deep Neural Networks

Deep neural networks have achieved great success on recommendation area. The most related works are



**Martin Scorsese**'s movie is close to the best **American** commercial cinema can get. The whole world is brought to our eyes. I am no fan of **Leonardo DiCaprio**, but he is at his best in this film. ......

Cold blooded murder. **Kate Beckinsale** stars as U.S.Marshall Carrie Stetko, who is dropped into a shocking mystery at a research station in **Antartica**. Also starring are: **Columbus Short**, **Gabriel Macht**, **Alex O'Loughlin** and a personal favorite of mine, **Tom Skerritt**. ......

Figure 1 The knowledge graph creation of our approach. The process contains three parts. The first part includes user reviews, item reviews and their rating information. The bold words are knowledge entities extracted from reviews. The second part is the fusion graph of reviews and ratings, the light blue nodes represent review entities, and the white nodes represent users and items. The third part is the final graph, and the dark blue nodes are extended from initialize entities to one hop away.

those review-based recommender systems [1], [2], [5], [8], [9], [12], [16]. They usually used two strategies to connect ratings and reviews. The first is to simply concatenate the rating embedding vector and the review embedding vector together, and then used a nonlinear layer to predict the final rating [2], [8]. Specifically, Wu et al. [8] built user-item graph based rating information, and used hierarchical neural networks to learn review features and graph rating features, and then concatenated them together as the final user/item representation. The second is to use nonlinear neural network to model the relationships between ratings and reviews [2], [5], [12]. For example, Peng et al. [5] proposed a neural gated fusion network to combine the review encoded features and rating encoded features together as the final representations, and then fed them into Latent Factor Model (LFM) for rating prediction. However, the above works cannot provide explainable recommendation results, since they cannot distinguish the specific contributions of ratings and reviews.

### 2 3 Knowledge-based Models

In order to alleviate the sparsity and cold-start problem in Recommender Systems (RS), some auxiliary information are introduced to help to improve recommendation performance, such as social information [6], [18], item content information [19] , reviews [17], and knowledge graph [11], [28], [20], [21], [22]. Knowledge graph is one of the widely used auxiliary information, which contains potential user preferences and items properties. Meanwhile, it has been confirmed that a knowledge graph can provide explainable recommendation results [11], [20]. The most direct way of using knowledge graph is to map items to knowledge entities [28] [11]. For example, Wang et al. [28] designed cross&compress units to model correlations between items and knowledge entities, and learned the potential properties of items in the knowledge graph. Another way is to identify entities from item contents like news topic [20], movie description [21]. Wang et al. [20] extracted knowledge entities from news topics, and learned news representation by using Knowledge Graph Embedding (KGE) method for news click prediction. Zhang et al. [21] modeled items from three knowledge aspects, structural knowledge, textual knowledge, and visual knowledge which extracted from item identifies, item summary, and item images, respectively. However, above works mainly modeled items properties based knowledge graph, and ignored the preference learning of users from knowledge spaces. To solve above problem, our work not only models item properties but also learns user preferences from a knowledge graph by extracting knowledge entities from user reviews.

## 3 Our Approach

In this section, we first explain how to build the knowledge graph by fusing ratings and reviews information. We then propose our RMK framework.

### 3 1 Fusion Knowledge Graph Construction

The process of fusion knowledge graph construction is illustrated in Figure 1. Given the user review documents and item review documents, we firstly extract knowledge entities from these documents by applying the entity linking method [23], [24]. Given user-item ratings, we regard users and items as nodes of the knowledge graph, and connect these nodes with entity nodes belonging to their reviews. In order to explore the potential user preferences and item prop-

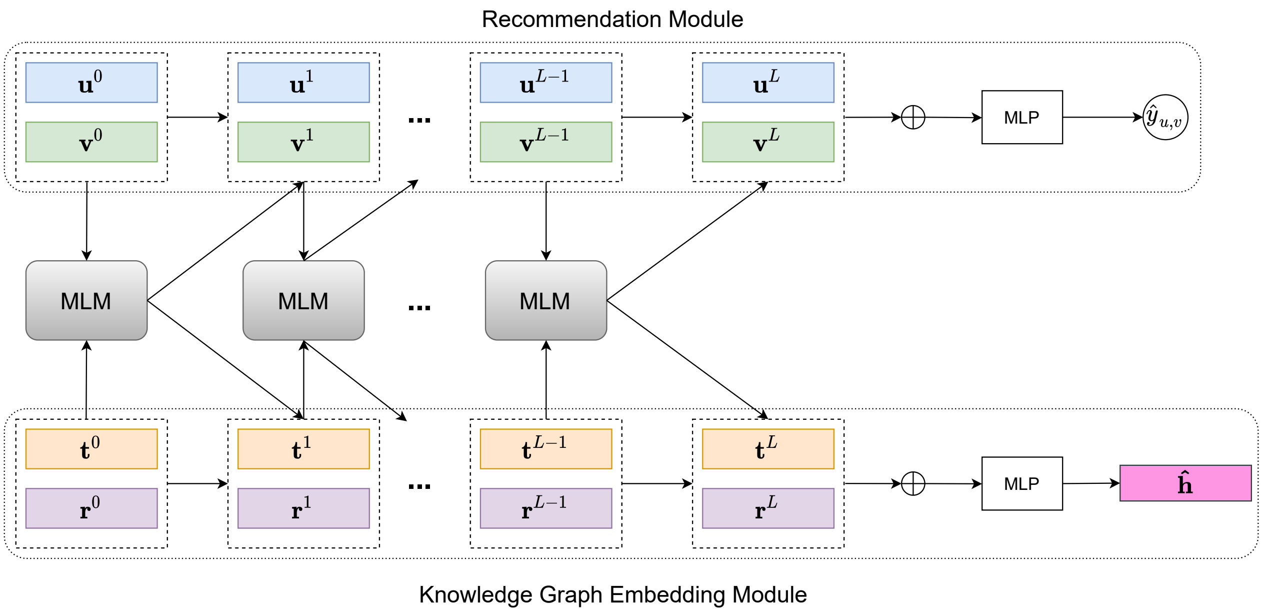


Figure 2 The Framework of RMK. Our model includes three main modules, the Recommendation Module, the Mutual Learning Module (MLM), and the Knowledge Graph Embedding Module. For Recommendation, users and items are treated as head nodes of KG, and put them into Mutual Learning Module (MLM) to learn the interaction between them with their review entities. For the KGE module, we use the tails and relations of KG as input to predict their head embeddings. The tails are treated as two-hop neighbors of users and items bridged by head $h$. We then learn the relations between the uses/items with their two-hop distance neighbors through MLM.

erties, we extend the connections of entities to one-hop away. One-hop in this paper is the distance that could connect entities that have a direct relationship with the current entity. Overall, the final knowledge graph contains users, items, review entities, extended entities, and relations between the above nodes.

We assume that a knowledge graph $\mathcal{G}$ can be formulated as a triple set, $\mathcal{G} = \{(h, r, t) \mid h, t \in \mathcal{E}, r \in \mathcal{R}\}$, where $\mathcal{E}$ and $\mathcal{R}$ represent the entities and relations, respectively. specifically, $h$ and $t$ denote the head entity and tail entity corresponding to the relation $r$, respectively. We define a set of users $U = \left\{u_1, u_2, ..., u_{|U|}\right\}$ and a set of items $V = \left\{v_1, v_2, ..., v_{|V|}\right\}$, where $|U|$ and $|V|$ represent the numbers of users and items, respectively. We denote the embedding vector for the $i$-th user as $\mathbf{u}_i$, where $i \in \{1, 2, ..., |U|\}$ and the embedding vector for the $j$-th item as $\mathbf{v}_j$, where $j \in \{1, 2, ..., |V|\}$.

### 3 2 RMK Framework

The framework of our model RMK is shown in Figure 2. Our model includes three parts, Recommendation Module, Mutual Learning Module (MLM), and Knowledge Graph Embedding (KGE) module. After building the knowledge graph, we learn the relations between users/items with their review entities and two-hop distance entities in KG through L layer MLM, and predict user $u$ rating $\hat{y}$ to item $v$. We learn entity embeddings by applying Knowledge Graph Embedding (KGE) [25] module. We adopt $L$ layers of KGE to learn entity embeddings, and select user embedding and item embeddings at each layer. Then, we use nonlinear neural networks to predict heads in KG based on their relation and tail. In summary, our framework is an end-to-end model with two mutual learning tasks. Besides, our model ensures that we run two tasks simultaneously.

### 3 3 Mutual Learning Module

Mutual Learning Module (MLM) is proposed for modeling the mutual relationship between the head entity $h$ and its neighbor entity set $N_h = \{h\langle 1\rangle, h\langle 2\rangle, ..., h\langle |N_h|\rangle\}$. The detailed design is shown in Figure 3.

In MLM, A $L$-layer linear projection model is used as the feature extractor. We denote $\mathbf{h}^l \in \mathbb{R}^d$ and $\mathbf{h}^l\langle i\rangle \in \mathbb{R}^d$ as the embedding vectors of the head entity and its $i$-th neighbor entity in the $l$-th layer, respectively. We build a $d \times d$ pairwise interactions of $\mathbf{h}^l$ and $\mathbf{h}^l\langle i\rangle$ as

$$\mathbf{C}^l = \mathbf{h}^l \left(\mathbf{h}^l\langle i\rangle\right)^\top = \begin{bmatrix} h_1^l h_1^l\langle i\rangle & \cdots & h_d^l h_1^l\langle i\rangle \\ \cdots & & \cdots \\ h_1^l h_d^l\langle i\rangle & \cdots & h_d^l h_d^l\langle i\rangle \end{bmatrix}, \quad (1)$$

where $\mathbf{C}^l \in \mathbb{R}^{d\times d}$ is the cross feature matrix of $\mathbf{h}^l$ and $\mathbf{h}^l\langle i\rangle$

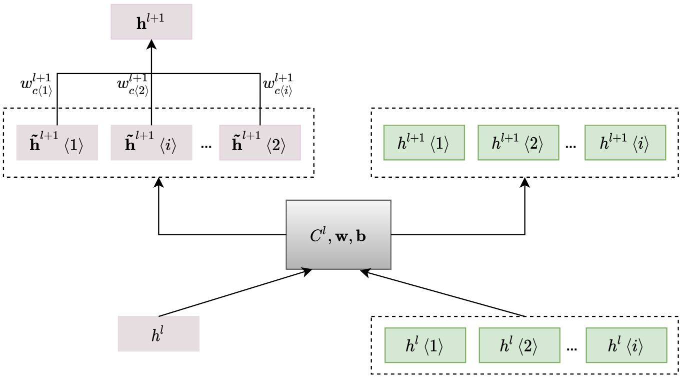

$\mathbf{h}^{l+1}$
$w^{l+1}_{c\langle 1\rangle}$
$w^{l+1}_{c\langle 2\rangle}$
$w^{l+1}_{c\langle i\rangle}$
$\tilde{\mathbf{h}}^{l+1}\langle 1\rangle$
$\tilde{\mathbf{h}}^{l+1}\langle i\rangle$
...
$\tilde{\mathbf{h}}^{l+1}\langle 2\rangle$
$h^{l+1}\langle 1\rangle$
$h^{l+1}\langle 2\rangle$
...
$h^{l+1}\langle i\rangle$
$C^l, \mathbf{w}, \mathbf{b}$
$h^l$
$h^l\langle 1\rangle$
$h^l\langle 2\rangle$
...
$h^l\langle i\rangle$

Figure 3 The Structure of Mutual Learning Module.

in the $l$-th layer. Then, we generate the representation vectors of the head entity and its $i$-th neighbor entity in the $l+1$ layer through mapping the $\mathbf{C}^l$ to the representation spaces as follows:

$$\begin{array}{l} \mathbf{h}^{l+1}\langle i\rangle = \mathbf{C}^l \mathbf{w}^l_{H^l\langle i\rangle H^l\langle i\rangle} + (\mathbf{C}^l)^\top \mathbf{w}^l_{H^l H^l\langle i\rangle} + \mathbf{b}^l_{H^l\langle i\rangle} \\ \mathbf{h}^{l+1} = \sum_{i=1}^{|N_h|} w^{l+1}_{c\langle i\rangle} \tilde{\mathbf{h}}^{l+1}\langle i\rangle + \mathbf{b}^{l+1}_c \end{array}, \tag{2}$$

where the vectors $\mathbf{w}^l_{**} \in \mathbb{R}^d$ and $\mathbf{b}^l_* \in \mathbb{R}^d$ denote the trainable projection weights and biases for the $l$-th layer, respectively. The symbol $*$ represents the entity variable. The number $w^{l+1}_{c\langle i\rangle}$ represents the contribution of the i-th neighbor to $h$ in the $l+1$ layer, and the vector $\mathbf{b}^{l+1}_c \in \mathbb{R}^d$ represent the biases corresponding the linear projection from the summed intermediate vector to the head entity in the $l+1$-layer. Note that $\tilde{\mathbf{h}}^{l+1}\langle i\rangle$ represents the intermediate vector corresponding to the head entity vector $h$ with $i$-th component in the $l+1$ layer, which can be calculated as follows:

$$\tilde{\mathbf{h}}^{l+1}\langle i\rangle = \mathbf{C}^l \mathbf{w}^l_{H^l\langle i\rangle H^l} + (\mathbf{C}^l)^\top \mathbf{w}^l_{H^l H^l} + \mathbf{b}^l_{H^l}. \tag{3}$$

By Eq.(1), Eq.(2) and Eq(3), our proposed MLM can capture the relationship between every pair of head entity and its every corresponding neighbor entity.

### 3 4 Recommendation Module

To specify the operations of extracting features from raw inputs in MLM. we formulate the extraction operations for user $u$ and item $v$ as

$$\begin{array}{l} \mathbf{u}^L = \mathbb{E}_{N_u\sim\mathcal{S}(u)}\left[C^L(\mathbf{u}, \mathbf{N_u})[\mathbf{u^{L-1}}]\right] \\ \mathbf{v}^L = \mathbb{E}_{N_v\sim\mathcal{S}(v)}\left[C^L(\mathbf{v}, \mathbf{N_v})[\mathbf{v^{L-1}}]\right] \end{array}, \tag{4}$$

where $L$ is the layer of MLM, $\mathbb{E}$ is sample operation, $\mathbf{N_u}$ and $\mathbf{N_v}$ denote the neighbor entities for each user and item. $S(u)$ and $S(v)$ represent the corresponding neighbor set of user $u$ and item $v$. The recommendation module is composed of a $G$-layer multi-layer perceptron (MLP). The final rating of the user $u$ to item $v$, $y_{u,v}$ can be predicted by projecting the concatenated vector of $\mathbf{u}^L$ and $\mathbf{v}^L$ as

$$\hat{y}_{u,v} = \underbrace{F\left(F\left(\ldots F\left([\mathbf{u}^L; \mathbf{v}^L]\right)\right)\right)}_{G-\text{layer MLP}}, \tag{5}$$

where $F(*)$ denotes the one-layer MLP projection.

### 3 5 Knowledge Graph Embedding Module

Knowledge Graph Embedding is to embed the KG triples into continuous vectors while preserving the structure of KG. Our KGE inherits the idea of transnational distance method [25]. In recommendation scenario, we mainly focus on the modeling of users and items, and our KGE module is not only used to learn one-hop distance entities but also two-hop distance entities of users and items. Specifically, for each $h$ in KG, we use $\mathbf{h}$ to represent the embedding vector of it and sample a user and an item linked to it. The user embedding and item embedding are denoted as $\mathbf{h}_u$ and $\mathbf{h}_v$. The matrix of neighbors corresponding to $h$ of $u$ is represented as $\mathbf{N}_{\mathbf{h}u}$. The matrix of neighbors corresponding to $h$ of $v$ is represented as $\mathbf{N}_{\mathbf{h}v}$.

$$\begin{array}{l} \mathbf{N}_{\mathbf{h}u}^{L} = \mathbb{E}_{h_u\sim\mathcal{S}(N_{h_u})}\left[C^L(\mathbf{h}_u, \mathbf{N}_{\mathbf{h}u})[\mathbf{N}_{\mathbf{h}u}^{L-1}]\right] \\ \mathbf{N}_{\mathbf{h}v}^{L} = \mathbb{E}_{h_v\sim\mathcal{S}(N_{h_v})}\left[C^L(\mathbf{h}_v, \mathbf{N}_{\mathbf{h}v})[\mathbf{N}_{\mathbf{h}v}^{L-1}]\right] \end{array}, \tag{6}$$

where L is the layer of MLM module, $\mathbb{E}$ is sample operation, $S(N_{h_u})$ is the user set of neighbor $N_{h_u}$, $S(N_{h_v})$ is the item set of neighbor of $N_{h_v}$.

$$\begin{array}{l} \mathbf{t}_h = \underbrace{F\left(F\left(\ldots F\left(\mathbf{N}_{\mathbf{h}u} \oplus \mathbf{N}_{\mathbf{h}v}\right)\right)\right)}_{P-\text{layer MLP}} \\ \mathbf{r}_h = \underbrace{F\left(F\left(\ldots F\left(\mathbf{r}\right)\right)\right)}_{P-\text{layer MLP}} \\ \hat{\mathbf{h}} = \underbrace{F\left(F\left(\cdots F\left(\begin{bmatrix} \mathbf{r}_h \\ \mathbf{t}_h \end{bmatrix}\right)\right)\right)}_{K-\text{layer MLP}} \end{array}, \tag{7}$$

where $\mathbf{t_h}$ is the tail embedding vector of $h$, $\mathbf{r_h}$ is the relation embedding vector of relation $r$ from head $h$, $\hat{\mathbf{h}}$ is the predicted head embedding of $h$. [] is a vertical concatenate operation.

$$\text{score}(h,r,t) = f_{KG}(\mathbf{h}, \hat{\mathbf{h}}), \tag{8}$$

where $\mathbf{h}$ is the real head embedding, $f_{KG}()$ is a sigmoid function.

### 3 6 Objective Function

The objective function of RMK is:

$$\begin{aligned}\mathcal{L} =& \mathcal{L}_{RS} + \mathcal{L}_{KGE} + \mathcal{L}_{REG} \\ =& \sum_{u\in U, v\in V} \mathcal{J}\left(\hat{y}_{u,v}, y_{u,v}\right) - \lambda_1 \sum_{(h,r,t)\in\mathcal{G}} \text{score}(h,r,t) \\ &+ \lambda_1 \sum_{(h',r,t')\notin\mathcal{G}} \text{score}\left(h',r,t'\right) + \lambda_2 \|\mathbf{W}\|_2^2\end{aligned}, \tag{9}$$

where $\mathcal{L}_{RS}$, $\mathcal{L}_{KGE}$ and $\mathcal{L}_{REG}$ are loss functions of recommendation module, knowledge graph embedding module, and regularization term, respectively. The symbol $\mathcal{J}(*)$ denotes the Mean Square Error (MSE) function. We use the $\lambda_1$ as the learning rate parameter for balancing the RS task and KGE task. $\lambda_2$ is the regularization parameter for the total objective function.

## 4 Experiments

In this section, we first describe the IMDb dataset used in the experiment. Then, we compare the performance of our proposed model, RMK, with baseline models.

### 4 1 Datasets

We conducted experiments with one publicly accessible datasets: the IMDb dataset. Table 1 lists the statistics of IMDb. This movie rating dataset was published as part of Jointly Modeling Aspects, Ratings and Sentiments (JMARS) [14], which uncovered aspects and sentiments from reviews to predict movie ratings. We preprocessed the dataset to ensure that each user had at least two reviews. The range of ratings for IMDb is $[0, 10]$.

Table 1 Basic statistics of IMDb dataset.

| statistics | IMDb |
|---|---|
| #users | 2,088 |
| # movies | 4,668 |
| # reviews | 126,874 |
| # entities | 1,219,228 |
| # distinct entities | 75,549 |

### 4 2 Baselines

We compared the proposed method, RMK, with the following state-of-the-art methods.

- Probabilistic Matrix Factorization (PMF). This method models the latent factors for users and items while ignoring all the review texts [26]. By comparing RMK with PMF, we evaluated the effectiveness of review-based recommendation.

Table 2 The comparison of RMSE on IMDb dataset.

| Method | RMSE |
|---|---|
| PMF | 1.916 |
| GMF | 1.865 |
| MPCN | 1.840 |
| MKR | 1.838 |
| **RMK** | **1.826** |

- Generalized Matrix Factorization (GMF). This method extends matrix factorization (MF) to a nonlinear model capturing the interaction between users and items. It learns a latent embedding from a user (item) identifier to represent a user (item) [27].
- Multi-Pointer Co-Attention Networks for Recommendation (MPCN). This is a review-by-review pointer-based model. It uses hard attention to extract important user and item reviews and subsequently matches them in a word-by-word fashion [17].
- Multi-task feature learning approach for Knowledge graph enhanced Recommendation (MKR). This method mapped item ids to knowledge entities and learned recommendation task and knowledge graph embedding task at the same time [28].

### 4 3 Evaluation Metric

To evaluate the performances of RMK and baselines, we adopted the well-known root-mean-square error (RMSE), which has widely been used in rating prediction of recommendation. Given a ground truth rating $y_{uv}$ and a predicted rating $\widehat{y}_{uv}$ for an interaction between user $u$ and item $v$, the RMSE is calculated as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{u,v}\left(\widehat{y}_{uv} - y_{uv}\right)^2}, \tag{10}$$

where $N$ indicates the number of user ratings for movies.

### 4 4 Experimental Result

The comparison results on IMDb datasets are shown in Table 2. The result shows that the proposed RMK outperforms other baseline models. The results of methods that only used rating information as features (PMF, GMF) are worse than those used reviews (MPCN, RMK). In addition, MKR and RMK with mutual learning design achieved the best performance. Our proposed method RMK not only used item entities but also considered review entities. The most important is we modeled the relations between entities and their neighbors, which is more effective than the MKR method.

## 5 Conclusions and Future Work

In this paper, we propose RMK, an end-to-end framework that fuses ratings and reviews in the same knowledge graph, and explores the correlations between them by running KGE and rating prediction tasks simultaneously. Since ratings and reviews are not independent but are connected by review entities and users/items, we treat users and items as nodes in the knowledge graph, just like review entity nodes. We design KGE task and rating prediction task and associate them based on the connections between users/items and entities. Furthermore, we learn the high-order interactions between ratings and reviews through L layer neural networks, and these two kinds of features can communicate at each layer.

For future work, we will conduct more experiments on other datasets, such as Amazon Movies and TV (注1) to evaluate our proposed model. We also focus on improving the effectiveness of mutual learning and provide further analysis on the contributions of different entity neighbors.

## Acknowledgement

This work was partly supported by JSPS KAKENHI Grant Numbers 18H03242, 18H03342, 19H01138A.

### References

[1] Lu Y, Dong R, Smyth B, “Coevolutionary recommendation model: Mutual learning between ratings and reviews,” [C]//Proceedings of the 2018 World Wide Web Conference. 2018: 773-782.

[2] Liu D, Li J, Du B, et al., “Daml: Dual attention mutual learning between ratings and reviews for item recommendation,” [C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019: 344-352.

[3] Alexandridis G, Tagaris T, Siolas G, et al., “From Free-text User Reviews to Product Recommendation using Paragraph Vectors and Matrix Factorization,”[C]//Companion Proceedings of The 2019 World Wide Web Conference. 2019: 335-343.

[4] McAuley J, Leskovec J., “Hidden factors and hidden topics: understanding rating dimensions with review text,”[C]//Proceedings of the 7th ACM conference on Recommender systems. 2013: 165-172.

[5] Peng Q, Liu H, Yu Y, et al., “Mutual Self Attention Recommendation with Gated Fusion Between Ratings and Reviews,”[C]//International Conference on Database Systems for Advanced Applications. Springer, Cham, 2020: 540-556.

[6] Li P, Lu H, Zheng G, et al., “Exploiting Ratings, Reviews and Relationships for Item Recommendations in Topic Based Social Networks,”[C]//The World Wide Web Conference. 2019: 995-1005.

[7] Tan Y, Zhang M, Liu Y, et al., “Rating-boosted latent topics: Understanding users and items with ratings and reviews, ”[C]//IJCAI. 2016, 16: 2640-2646.

[8] Wu C, Wu F, Qi T, et al., “Reviews meet graphs: Enhancing user and item representations for recommendation with hierarchical attentive graph neural network,”[C]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019: 4886-4895.

[9] Chen W, Zheng H T, Wang Y, et al., “Utilizing Generative Adversarial Networks for Recommendation based on Ratings and Reviews,”[C]//2019 International Joint Conference on Neural Networks (IJCNN). IEEE, 2019: 1-8.

[10] Wang H, Zhang F, Zhao M, et al., “Multi-task feature learning for knowledge graph enhanced recommendation,”[C]//The World Wide Web Conference. 2019: 2000-2010.

[11] Wang H, Zhang F, Wang J, et al., “Ripplenet: Propagating user preferences on the knowledge graph for recommender systems,”[C]//Proceedings of the 27th ACM International Conference on Information and Knowledge Management. 2018: 417-426.

[12] Chen C, Zhang M, Liu Y, et al., “Neural attentional rating regression with review-level explanations,”[C]//Proceedings of the 2018 World Wide Web Conference. 2018: 1583-1592.

[13] Ling G, Lyu M R, King I., “Ratings meet reviews, a combined approach to recommend,”[C]//Proceedings of the 8th ACM Conference on Recommender systems. 2014: 105-112.

[14] Diao Q, Qiu M, Wu C Y, et al., “Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS),“[C]//Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. 2014: 193-202.

[15] Peña F J, O’Reilly-Morgan D, Tragos E Z, et al., “Combining Rating and Review Data by Initializing Latent Factor Models with Topic Models for Top-N Recommendation,“[C]//Fourteenth ACM Conference on Recommender Systems. 2020: 438-443.

[16] Catherine R, Cohen W., “Transnets: Learning to transform for recommendation,“[C]//Proceedings of the eleventh ACM conference on recommender systems. 2017: 288-296.

[17] Tay Y, Luu A T, Hui S C., “Multi-pointer co-attention networks for recommendation,“[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2018: 2309-2318.

[18] Zhao T, McAuley J, King I., “Leveraging social connections to improve personalized ranking for collaborative filtering,“[C]//Proceedings of the 23rd ACM international conference on conference on information and knowledge management. 2014: 261-270.

[19] Lu Z, Dou Z, Lian J, et al., “Content-based collaborative filtering for news topic recommendation,“[C]//AAAI. 2015, 15: 217-223.

[20] Wang H, Zhang F, Xie X, et al., “DKN: Deep knowledge-aware network for news recommendation,“[C]//Proceedings of the 2018 world wide web conference. 2018: 1835-1844.

[21] Zhang F, Yuan N J, Lian D, et al., “Collaborative knowledge base embedding for recommender systems,“[C]//Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. 2016: 353-362.

[22] Zhao H, Yao Q, Li J, et al., “Meta-graph based recommendation fusion over heterogeneous information networks,“[C]//Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017: 635-644.

[23] Cheng X, Roth D., “Relational inference for wikification,“[C]//Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. 2013: 1787-1796.

[24] Ratinov L, Roth D, Downey D, et al., “Local and global algorithms for disambiguation to wikipedia,“[C]//Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies. 2011: 1375-1384.

[25] Bordes A, Usunier N, Garcia-Duran A, et al., “Translating embeddings for modeling multi-relational data,“[J]. Advances in neural information processing systems, 2013, 26: 2787-2795.

[26] Mnih A, Salakhutdinov R R. Probabilistic matrix factorization[J]. Advances in neural information processing systems,

(注1)：http://jmcauley.ucsd.edu/data/amazon/

2007, 20: 1257-1264.

[27] He X, Liao L, Zhang H, et al. Neural collaborative filtering[C]//Proceedings of the 26th international conference on world wide web. 2017: 173-182.

[28] Wang H, Zhang F, Zhao M, et al. Multi-task feature learning for knowledge graph enhanced recommendation[C]//The World Wide Web Conference. 2019: 2000-2010.