Engineering Applications of Artificial Intelligence 128 (2024) 107499

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Keyword-enhanced recommender system based on inductive graph matrix completion

Donghee Han [a], Daehee Kim [a], Keejun Han [b,*], Mun Yong Yi [c,*]

[a] Graduate School of Data Science, KAIST, Daejeon 34141, Republic of Korea
[b] Division of Computer Engineering, Hansung University, Seoul 02876, Republic of Korea
[c] Department of Industrial and Systems Engineering, KAIST, Daejeon 34141, Republic of Korea

ARTICLE INFO

Keywords:
Graph neural network
Inductive graph matrix completion
Keyword-enhanced subgraph
Transferability
Recommender system

ABSTRACT

Going beyond the user–item rating information, recent studies have utilized additional information to improve the performance of recommender systems. Graph neural network (GNN) based approaches are among the most common. However, existing models that utilize text data require a lot of computing resources and have a complex structure that makes them difficult to utilize in real-world applications. In this research, we propose a new method, keyword-enhanced graph matrix completion (KGMC), which utilizes keyword sharing relationships in user–item graphs. Our model has a simpler structure and requires less computing resources than existing models that utilize text data, but it has the advantage of cross-domain transferability while providing an intuitive understanding of the inference results. KGMC consists of three steps: (1) keyword extraction from the review text, (2) subgraph extraction and keyword-enhanced subgraph construction, and (3) GNN-based rating prediction. We have conducted extensive experiments over eight benchmark datasets to examine the relative superiority of the proposed KGMC method, compared to state-of-the-art baselines. Additional experiments and case studies have been also conducted to demonstrate the transferability as well as keyword-based explainability of KGMC. Our findings highlight the practical advantages of our model for recommender systems and support its effectiveness in inductive graph-based link prediction.

1. Introduction

As the number of choices in our daily lives is rapidly increasing, matching potentially relevant items to users is becoming increasingly important. In particular, the recent development of artificial intelligence technologies has rekindled the need to develop highly sophisticated recommender systems. Deep learning methods have enabled multi-modal data such as texts and images, to be better utilized in recommender systems (Zhang et al., 2019, 2022; Sharma et al., 2023; Valcarce et al., 2019).

In recent years, graph neural networks (GNNs) have emerged as a new representation approach to enable multi-modal learning for recommender systems (Ying et al., 2018). Relative to other deep-learning approaches, GNNs can model the user–item relationships in a more intrinsic way (Wu et al., 2022a). While each GNN layer propagates node features, additional features can be smoothly utilized by inserting new nodes on an existing graph, allowing the potential for GNNs to be exploited for multi-modal learning (Shi et al., 2023). For GNN multi-modal learning approaches, individual embedding generation models such as VGG (Simonyan and Zisserman, 2015) and Word2Vec (Mikolov et al., 2013) have been adapted to the type of input (i.e., text or image). Then, those embeddings have been aggregated to represent node features (Ying et al., 2018). To extract more latent relationships from user–item interactions, side information such as reviews, timestamps, and item attributes have also been actively utilized (Gao et al., 2020). These approaches, however, require excessive computing resources because they need to utilize models with many parameters.

Recently, Transformer-based language models and convolutional neural network (CNN)-based image models have become popular for utilizing multi-modal data (Radford et al., 2021; Xu et al., 2023; Gajbhiye and Nandedkar, 2022; Parvaiz et al., 2023), but they require a great amount of GPU resources. Furthermore, adding embedding generation models to the recommender system to utilize multi-modal data also increases the overall system and infrastructure complexity. Conversely, existing multi-modal approaches utilize text and images converted to embeddings, making it difficult for humans to intuitively understand the results. In the context of e-commerce, monitoring is important to prevent recommender systems from performing poorly over time due to problems such as filter bubbles (Jiang et al., 2019) or echo chambers (Ge et al., 2020).

* Corresponding authors.
E-mail addresses: handonghee@kaist.ac.kr (D. Han), dh.kim@kaist.ac.kr (D. Kim), keejun.han@hansung.ac.kr (K. Han), munyi@kaist.ac.kr (M.Y. Yi).

https://doi.org/10.1016/j.engappai.2023.107499
Received 25 June 2023; Received in revised form 29 October 2023; Accepted 10 November 2023
Available online 15 November 2023
0952-1976/

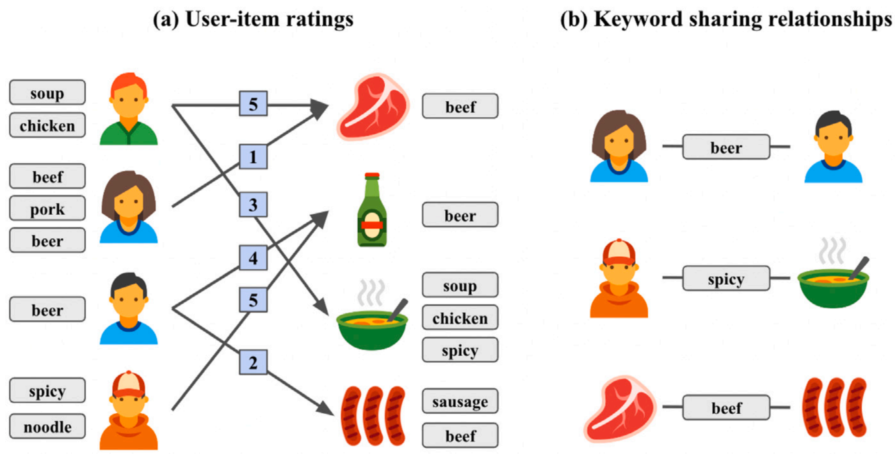

Fig. 1. Utilizing keyword sharing relationships in addition to user–item ratings.

Another useful property that a recommender system needs is transferability; the ability to leverage data from other domains to operate a recommender system is important if sufficient training data is not available in the early stages. Undoubtedly, transferable models have been widely used in computer vision and natural language processing, with some significant progress in the GNN field (Zhu et al., 2021; Han et al., 2021). However, transferable GNNs have only focused on learning node features from the structure of the graph. User reviews are highly valuable information, and enabling the transformers to be review-aware may further enhance GNN-based recommender systems. However, when user reviews are transferred to multi-dimensional vectors, it is difficult for them to be used with the graph structure simultaneously; thus, some studies have attempted to utilize user reviews in a message-passing step, requiring additional execution time and resources as they need additional graph attention layers (Gao et al., 2020). As the architecture does not fully support transferability for a new domain, additional information such as user reviews should be used as graph components when building a graph.

In this study, we focus on capturing additional information from the text to improve the quality of recommendations and propose a keyword-enhanced graph matrix completion (KGMC) method, which utilizes keywords of users and items in an inductive graph-based rating prediction model. The proposed model generates documents based on user reviews and uses keywords extracted from the documents as side information. Specifically, the target user, items, and their neighbors are extracted as a subgraph, and keyword co-occurrences between nodes are added as additional edges into the subgraph. Additional edges are categorized into three types: user–user, item–item, and user–item, and each type is learned in different message passage layers. By doing so, the proposed model also becomes explainable as those keywords can serve as explicit information in understanding the prediction results. Also, unlike existing models, our model does not convert text data into embeddings, resulting in a simpler and lighter model structure. These advantages reduce operational risk in real-world applications. Another important advantage of our model is transferability. As we mentioned earlier, our model learns additional relationships between nodes by converting keywords into edges in the graph, enabling transfer learning based on graph structure. This transferability is useful in e-commerce environments where there are multiple categories.

Fig. 1 shows the use of keyword-sharing relationships in addition to user–item ratings. Shared keywords between user–user, item–item, and user–item can be represented as additional edges, which can reveal additional relationships that are traditionally difficult to discover using ratings alone. In particular, relationships between homogeneous nodes are not found in rating-based relationships. In e-commerce, similar items may be registered with different IDs, such as when a renewed product is released, and it is difficult for existing rating-based models to capture the similarity between two items (Liu et al., 2020). The method we propose can solve these problems through additional keyword sharing relationships, while at the same time providing an intuitive interpretation of the results.

In short, the main contributions of our study are as follows:

- We propose a transferable keyword-enhanced graph matrix completion (KGMC) model that utilizes keywords as relationships between nodes.
- We show that the proposed model's outcomes are explainable by investigating keywords in target subgraphs.
- We show that the proposed model outperforms other GNN-based models and review-aware models by conducting extensive experiments on various cross-domain datasets.

The remainder of this paper is structured as follows. Section 2 reviews related studies and Section 3 provides the overview and detailed procedures of KGMC. Section 4 demonstrates the effectiveness of the KGMC using multiple datasets. Lastly, Section 5 concludes the study with key highlights.

2. Related work

This section reviews several relevant studies on exploiting GNN-based recommender models and review-based recommendation models, followed by some prominent studies on keyword-extraction methods.

2.1. GNN-based recommender models

GNNs have been widely used in recommender systems and various studies have been conducted to apply them to large-scale data. Currently, the inductive method is considered the mainstream in GNN-based recommender systems (Wu et al., 2022a), as it produces high performance through subgraph learning (Ying et al., 2018; Hamilton et al., 2017; Zhang and Chen, 2020) and can infer unseen cases as well as achieve scalability to large graphs.

GraphSAGE (Hamilton et al., 2017) is a generic inductive model that extracts subgraphs through a k-hop random walk from each node. In each subgraph, a representation of the central node is obtained through graph convolutional networks. In this way, the model learns the topological structure of each subgraph as well as the neighboring node features. Given the advantages, new models that seek to further extend the original work have been proposed (Ying et al., 2018; Afoudi et al., 2023). For example, PinSAGE (Ying et al., 2018) is a model that extends GraphSAGE for multimodal data collected from Pinterest.[1] The Pinterest dataset contains images, texts, and connections between them. Each node feature is given by concatenating an image vector and a text vector.

IGMC (Zhang and Chen, 2020) is a model that applies an inductive approach to rating prediction using a rating matrix. While existing inductive methods extract subgraphs based on a single node, IGMC samples a subgraph from a user–item bipartite graph based on a user–item pair and their k-hop neighbors. IGMC achieves higher performance by using only graph structure information without any node features. Another advantage of IGMC is transferability; training large graphs demands a large amount of resources, making transferability an important requirement.

EGI (Zhu et al., 2021) is an unsupervised model that learns subgraphs that sample k-hop neighbors around the target node. The model consists of a GNN encoder and decoder, and the central node embeddings are obtained from each subgraph. In Zhu et al. (2021), the transferability of the model was demonstrated and it was shown that transferability is based on the structural similarity of graphs. In addition, to measure the transferability between different graphs, they proposed an EGI gap that compares structural similarity, arguing that transfer learning in GNNs depends on graph structure information.

Heterogeneous GNNs have been proposed to more effectively learn heterogeneous graphs composed of various types of nodes and edges (Bing et al., 2023). Heterogeneous Graph Transformer (HGT) (Hu et al., 2020) is a representative heterogeneous GNN model that utilizes queries from target nodes and generates keys and values from neighbors. It also reflects heterogeneity by using different weight metrics for each node and edge type. SeHGNN (Yang et al., 2023) simplifies complexity by pre-computing neighbor aggregation using a lightweight mean aggregator to capture structural information. This method enhances semantic understanding through a single-layer structure with an extended receptive field via long metapaths and a transformer-based fusion module for combining features from diverse metapaths.

Recently, contrastive learning has emerged as an important research stream in the GNN field to address data sparsity and noise (Xie et al., 2023). NCL (Lin et al., 2022) takes into account the potential influence of neighboring nodes in constructing contrastive pairs. This model considers both the structural neighbors obtained from the interaction graph and the semantic neighbors based on the semantic space. SimGCL (Yu et al., 2022) eliminates the need for intricate graph augmentations and instead introduces uniform noise into the embedding space to create contrasting views. This method not only simplifies the process but also demonstrates improved recommendation accuracy and more efficient training compared to traditional graph augmentation-based methods. XSimGCL (Yu et al., 2023) is an extremely streamlined variant of SimGCL, in which the concurrent GNN layer dedicated to a contrasting task is eliminated. Instead, node embeddings are derived by a sequence of perturbated GNN layers. The contrastive task is executed by assessing the cross-layer distinctions between the final node embeddings and the node embeddings form the first GNN layers. Combining the two streams of research on heterogeneous GNN and contrastive learning, HGCL (Chen et al., 2023) utilizes diverse relationships in heterogeneous graph data through contrastive learning, improving user–item interaction modeling. It incorporates meta-networks for personalized contrastive augmentation, leading to enhanced recommendation performance on real-world datasets.

2.2. Review-based recommendation models

Another use for the recommender system is to extract and utilize the potential relationships between users and items by employing user reviews in text. As users' reviews have been accumulated in various web services, significant research efforts have been made to utilize them through natural language processing techniques.

TopicMF (Bao et al., 2014) is a topic-aware rating prediction model. This model utilizes topics in reviews along with the user–item rating matrix in conjunction with the matrix factorization method.

DeepCoNN (Zheng et al., 2017) is a CNN-based review-aware model that generates item and user features from review text; two parallel CNN layers generate user and item embeddings from a set of reviews, respectively.

NARRE (Chen et al., 2018) is a model that learns the importance of individual reviews through neural attention regression on two parallel CNN layers that generate user and item embeddings. In this way, the model judges the usefulness of each review and improves its explainability.

Recently, there have been attempts to use review text as additional information in graph-based models. AGCR (Kumar, 2022) learns review text through CNN and predicts the rating by using the review embeddings in a graph convolutional network (GCN) based model. This method constructs a user–item bipartite graph including ratings and reviews, and creates node embeddings using the attention mechanism for each domain of the rating and review.

RGCL (Shuai et al., 2022) is a review-aware graph contrastive learning model that utilizes review text through BERT-Whitening. The review vectors are employed as edge features in the user–item graph. This graph is then processed using a GCN to generate representations of both the user and item nodes. Furthermore, higher performance was achieved by utilizing contrastive learning to better distinguish different types of nodes and edges.

MEGCF (Liu et al., 2022) utilizes entities extracted from multimodal data such as images and text by adding them to the user–item graph as nodes. MEGCF employs GNN layers to capture intricate semantic relationships and collaborative filtering signals, while sentiment information from reviews further refines the recommendation process.

AHOR (Wang et al., 2023) predicts user–item links by incorporating not only user and item relationships, but also their respective aspects into the graph. The aspects of each user and item are obtained through diverse methods such as topic analysis from review text.

Although these review-based models have achieved high performance, no performance comparison with inductive models such as IGMC has been made. Also, the models using the review text in the GNN have relied on the method of converting the review text into vectors and utilizing them as edge features. The model we propose is different in that it uses the review text as additional graph structural information based on keywords rather than vectors. In this way, the model achieves high transferability and explainability.

2.3. Keyword-extraction methods

TF-IDF (Sammut and Webb, 2010) is one of the most well-known keyword extraction methods. It leverages *term frequency* (TF) and *inverse document frequency* (IDF) scores to select words that best represent a document. TF measures term occurrences in a document, favoring frequent terms. Conversely, IDF gauges how rarely a keyword appears across documents, valuing unique terms. The product of TF and IDF prioritizes frequent and distinctive terms for document representation. However, calculating IDF requires multiple documents, making it unsuitable for single documents.

TextRank (Mihalcea and Tarau, 2004) is an algorithm that uses PageRank (Gleich, 2015), an algorithm that calculates the rank of nodes in a graph. In TextRank, an edge between words is created and a graph is constructed using the co-occurrence relationships of words. The constructed graph calculates the rank of each word through the PageRank algorithm and extracts the top-ranked words as keywords.

KeyBERT (Grootendorst, 2020) is a keyword extraction method based on the BERT model (Devlin et al., 2019), which has been widely

[1] https://www.pinterest.co.kr/.

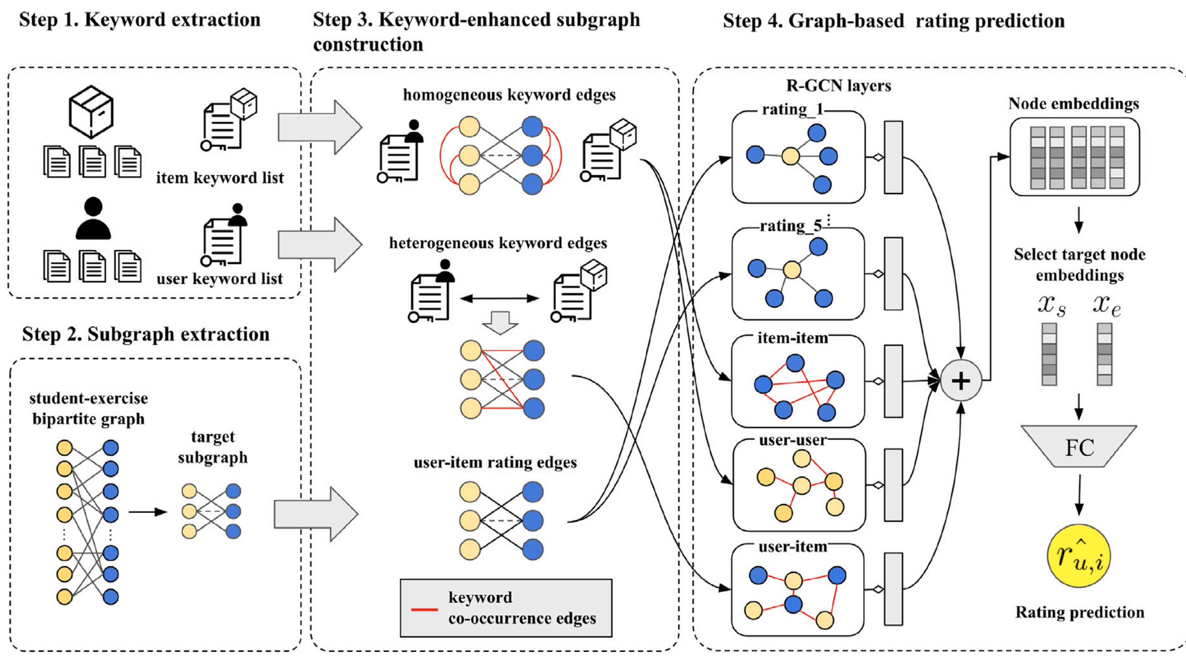

Fig. 2. The overall framework of the proposed KGMC (keyword-enhanced graph matrix completion) method.

used in recent studies in the NLP field. The previous two keyword extraction methods do not consider the meaning of words and sentences. In contrast, KeyBERT uses a pre-trained language model to compare the meanings of candidate words and sentences, and selects words with high similarity to the relevant document as keywords.

In our study, we compare the performances of inductive graph-based methods using all of the three aforementioned keyword extraction methods. Instead of evaluating each keyword extraction method at the item and user document level, a suitable keyword extraction method is determined based on the contribution to rating prediction performance improvement.

Algorithm 1 KGMC training algorithm

Input: Graph $\mathcal{G}(V, E, R)$; Keyword set K;
Output: GNN model Ψ;
for each minibatch **do**
  Extract $\mathcal{G}_s(V_s, E_s, R_s)$ from $\mathcal{G}(V, E, R)$
  Generate keyword co-occurrence edges E_{kw} from subgraph $\mathcal{G}_s$ and keyword set K
  $\hat{r} \leftarrow \Psi(\mathcal{G}_s(V_s, E_s + E_{kw}, R_s))$
  Backward propagation $L(r, \hat{r})$
end for

3. Method

This section explains the structure and the training process of the proposed method. As shown in Fig. 2, KGMC consists of four main steps. First, user and item documents are constructed from reviews and keywords are extracted from each document (keyword extraction). Second, a user–item bipartite graph is constructed and subgraphs are extracted based on the target user–item node and their neighbors (subgraph extraction). Third, keyword co-occurrence edges are added to the subgraph (keyword-enhanced subgraph construction). Finally, node embeddings are obtained from R-GCN layers (graph-based rating prediction). In this final step, the target user–item rating is obtained from a fully connected layer. In the subsequent subsections, each step is described in detail. In addition, Algorithm 1 describes the KGMC training process where $\mathcal{G}$ is the entire user–item bipartite graph and $\mathcal{G}_s$ is the target subgraph. R and K denote the ratings and keywords set in graphs, respectively. Keyword co-occurrence edges E_{kw} are added to $\mathcal{G}_s$ and a GNN-based rating prediction model Ψ predicts rating $\hat{r}$ from the subgraph $\mathcal{G}_s$.

3.1. Keyword extraction

For keyword extraction, user and item documents are first constructed. The proposed method KGMC uses keywords, instead of embedding vectors, obtained from the review text, differentiating itself from existing review-based models. Fig. 3 describes the keyword extraction process. To extract keywords representing each user and each item, user documents and item documents are constructed from reviews, respectively. The figure shows four reviews written on an item. From them, an item document is constructed and a set of keywords is extracted by one of the three keyword extraction methods. User keywords are extracted using the same processes.

We extract n keywords from each document. TF-IDF, TextRank, and KeyBERT are used as keyword extraction techniques. TF-IDF extracts keywords considering the frequency of appearance in other documents through the IDF index (Sammut and Webb, 2010), whereas TextRank and KeyBERT extract keywords while receiving a single document as the input. TF-IDF can consider the appearance of words globally. However, given that only the frequency of words is considered, it is difficult to reflect the meanings and contexts of the words in the document. In contrast, KeyBERT can consider the sentiments and contexts of words. KeyBERT and TextRank are more efficient for a web application environment where documents are gradually accumulated because keyword extraction is required only for the added documents.

3.2. Subgraph extraction

A user–item bipartite graph constructed from the user–item interaction contains rating information. Each node represents a user or an item, and each edge represents the rating left by the user on the item. Similar to the IGMC, target users, item nodes, and their neighbors are extracted as subgraphs. The entire graph consists of two nodes: user and item. The ratings between users and items are utilized as edges. The rating is divided into five levels, each with a different type of edge. The

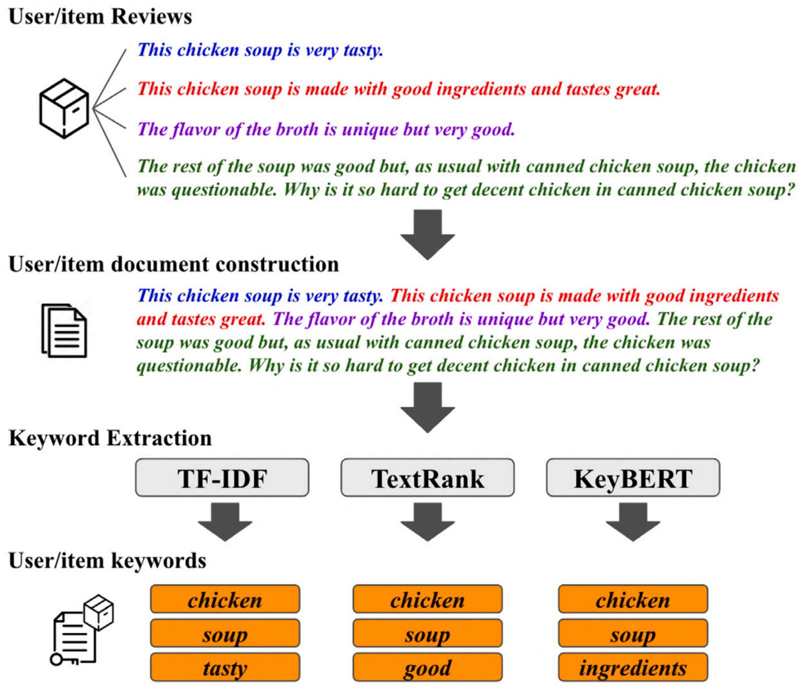

Fig. 3. User/item document construction & keyword extraction.

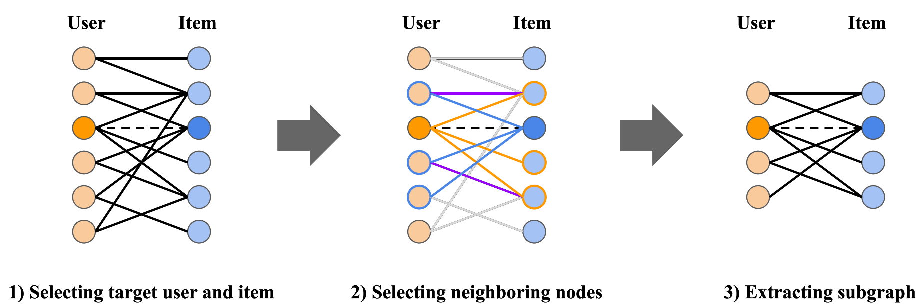

Fig. 4. Subgraph extraction process. Dash lines represent target interactions, and in the second step, orange lines represent interactions with the target user's 1-hop neighbors, blue lines represent interactions with the target item's 1-hop neighbors, and purple lines represent interactions between neighboring nodes.

subgraph contains four types of nodes: target user, target item, target user's neighbors, and target item's neighbors. The edges connecting the chosen nodes are included in the subgraph. To maintain the size of the subgraph below a certain scale, the number of neighboring nodes is limited. Fig. 4 shows the subgraph extraction process. In the first step, the target user and item nodes are selected. In the second step, the neighbors that are directly connected to the two target nodes are selected. The selected neighbors are marked with orange and blue outlines. Finally, the subgraph including the edges between the selected nodes is extracted.

To initialize the node vectors in the subgraph, the target user and item labels are set to 0 and 1, respectively. The labels for the neighboring users are set to 2 and those for the neighboring items are set to 3. For each node, one hot label vector is assigned as the initial node feature. These node features contain only graph structure information without any information about the user or item. This minimal requirement is very useful in cases where user and item information is not available, and it is also useful from a transfer learning perspective because it can be applied to cases where the user and item information have different schemas depending on the domain. Fig. 5 shows the node feature initialization process.

3.3. Keyword-enhanced subgraph construction

Edges are added to connect items and users that share common keywords, based on their co-occurrences. Algorithm 2 presents the pseudocode for the keyword-enhanced subgraph construction. k_u denotes keywords representing node u and an additional edge connecting two nodes is added to subgraph $\mathcal{G}_s$ when two nodes have one or more common keywords. Three types of keyword edges are created according to the type of node pair. A heterogeneous edge connects users and items and a homogeneous edge connects nodes of the same type, such as user–user and item–item. e_u, e_i, and e_{ui} denote keyword co-occurrence edges between user–user, item–item, and user–item, respectively. The subgraph to which the keyword co-occurrence edges are added is named the keyword-enhanced subgraph.

Fig. 6 shows the keyword co-occurrence edge generation process. In the example shown in the figure, for a homogeneous keyword edge, the edge is created because the u_1-u_2 node pair shares keyword B, and the i_1-i_3 node pair shares keyword D. For a heterogeneous keyword edge, the edge is created because the u_2-i_3 node pair shares keyword B, and for the u_3-i_2 node pair, the edge is created because they share keyword A.

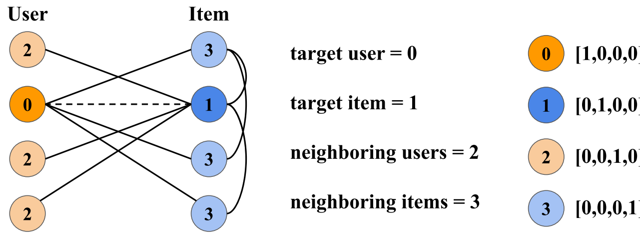

Fig. 5. Node feature initialization process.

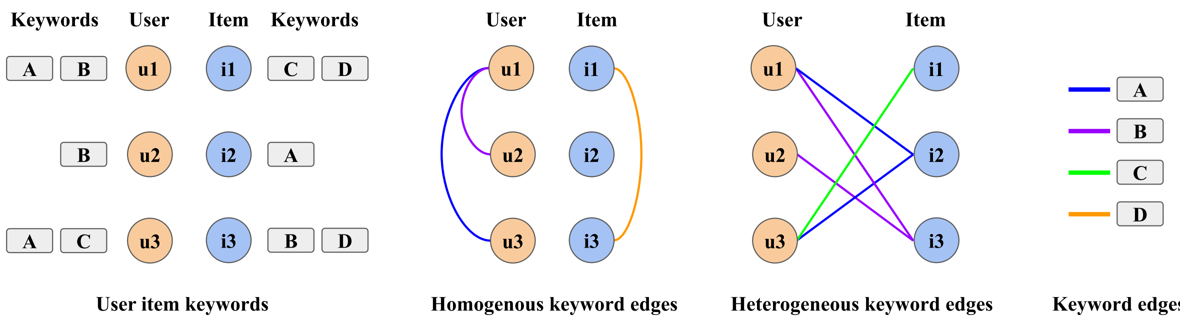

Fig. 6. Keyword co-occurrence edge generation process.

Algorithm 2 Keyword-enhanced subgraph construction algorithm

Input: User-item rating bipartite graph $\mathcal{G}$; Target user ID uid_t; Target item ID iid_t; Keyword set K;
Output: Target subgraph g;
/* Extract subgraph & add keyword edges */
initialize node set V
add uid_t, iid_t to V
$N_u \leftarrow$ get 1-hop neighbor item node set of uid_t from $\mathcal{G}$
$N_i \leftarrow$ get 1-hop neighbor user node set of iid_t from $\mathcal{G}$
$V \leftarrow merge(V, N_i, N_u)$
$\mathcal{G}_s \leftarrow$ extract subgraph from $\mathcal{G}$ with V
$P \leftarrow combination(V, 2)$
for u, v in P **do** $k_u, k_v \leftarrow K(u), K(v)$ /* get keywords of node u and v */
  if keyword co-occurrence in k_u, k_v **then** /* check u and v share same keywords */
    if u, v are all user node **then**
      $\mathcal{G}_s(u, v) \leftarrow e_u$
    else if u, v are all item node **then**
      $\mathcal{G}_s(u, v) \leftarrow e_i$
    else
      $\mathcal{G}_s(u, v) \leftarrow e_{ui}$
    end if
  end if
end for
Return g

By combining the two types of edges, three variants are proposed. In a prior study, it was shown that augmenting the inter-class, edge-connecting nodes of the same type is effective in GNN (Zhao et al., 2021). Extending the finding, we analyze whether there is a difference in performance among the three variants. The first variant (KGMC-hetero) is a model that uses only one heterogeneous edge and only keyword co-occurrence information between the user and item. The second variant (KGMC-homo) is a homogeneous type that uses only keyword co-occurrence information between nodes of the same type and creates an edge that connects user–user and item–item. User–user and item–item edges are divided into two different edge types. The last variant (KGMC-mixed) is a mixed type that uses both of the two edge types, heterogeneous and homogeneous. Fig. 7 shows how these variants are different in adding additional edges, relative to IGMC.

3.4. Graph-based rating prediction

In this step, the constructed keyword-enhanced subgraph in the previous step is passed through the relational graph convolutional network (R-GCN) layer to generate node representations. First, the relationships between items and users are represented as shown in Eq. (1),

$$\begin{aligned} R_r &= [\,1,\ 2,\ 3,\ 4,\ 5\,] \\ R_k &= \left[\, e_{ui},\ e_{uu},\ e_{ii} \,\right] \\ R &= R_r \cup R_k\,, \end{aligned} \tag{1}$$

where R_r refers to the set of five rating types and R_k refers to the set of three keyword co-occurrence edge types in which e_u, e_i and e_{ui} denote keyword co-occurrence edges between user–user, item–item, and user–item, respectively. The two edge type sets are combined as R, which is used to pass messages at each layer. Five rating types are used because most datasets contain five-level ratings. However, depending on the dataset, various numbers of rating levels and edge-type layers can be used.

As shown in Eq. (2), R-GCN performs message passing by classifying layers according to the edge types as

$$x_i^{(l+1)} = tanh\left(W_0^{(l)} x_i^{(l)} + \sum_{r \in R} \sum_{j \in \mathcal{N}_i^r} W_r^{(l)} x_j^{(l)} \right), \tag{2}$$

where $x_i^{(l+1)}$ is a representation vector of node i obtained from the lth R-GCN layer. In the equation, $\mathcal{N}_i^r$ is a set of neighbors connected to node i by edge type r, x_j is a feature of node j, and W_r is the weight for the linear transformation for each relation. We use the $tanh$ as the activation function.

Eq. (3) shows the process of predicting the final rating from node representation as

$$\begin{aligned} x_i' &= concat(x_i^1, x_i^2, \ldots, x_i^L) \\ h &= concat(x_s', x_e') \\ \hat{r} &= \sigma(W h + b)\,. \end{aligned} \tag{3}$$

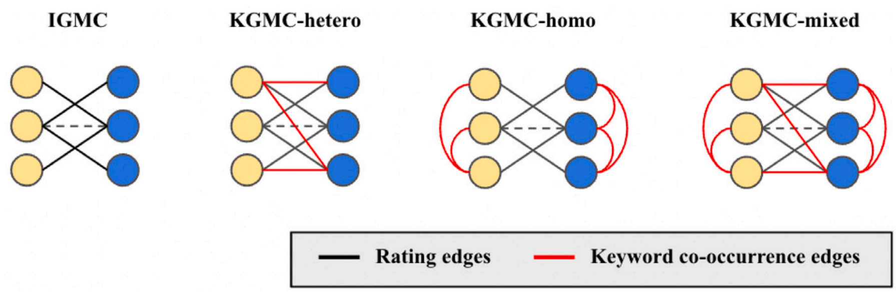

Fig. 7. Comparison of additional edge addition methods.

Table 1
Basic statistics of eight datasets.

Dataset	Movie	Music	Game	Sport	Office	Grocery	Yelp	Epinions
#Reviews	52,326	125,454	252,518	385,689	399,555	1,143,470	484,211	25,442
#Users	7559	19,498	31,603	60,504	57,374	127,487	40,610	5654
#Items	5744	21,155	25,501	63,511	43,883	41,320	27,668	4380
#Reviews per User	6.92	6.43	7.99	6.96	6.37	8.97	11.92	4.50
#Reviews per Item	9.11	5.93	9.90	9.11	6.07	27.67	17.50	5.81
#Words per Review	33.37	20.75	56.19	31.12	27.07	37.37	106.60	17.77
Density	0.121%	0.030%	0.031%	0.010%	0.016%	0.02%	0.04%	0.10%

Table 2
Comparison of baselines.

Component	NMF	PMF	SVD++	DeepCoNN	NARRE	IGMC	SimGCL	XSimGCL	RGCL	MEGCF	AHOR	**KGMC**
MF	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
CNN	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗
BERT	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓
GNN	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓
Rating	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓
Review	✗	✗	✗	✓	✓	✗	✗	✗	✓	✓	✓	✓

The final node representation x_i' is obtained by concatenating the representations x_i^l of node i obtained from each R-GCN layer l. The node representations x_u and x_i of the target user and item are extracted through different poolings and then concatenated into a single vector. The concatenated vector h is subjected to a fully-connected layer and a sigmoid function σ to obtain a predicted rating $\hat{r}$.

The mean square error (MSE) is used as a loss function. Eq. (4) shows how MSE loss is calculated:

$$\mathcal{L}_{MSE} = \frac{1}{N} \sum_{i=1}^{n} (r_i - \hat{r}_i)^2 , \tag{4}$$

where r_i and $\hat{r}_i$ denote the actual and predicted ratings of the user–item interaction i, respectively, and N denotes the total number of interactions in the training dataset. Adam optimizer (Kingma and Ba, 2015) is also used for training.

4. Experiment and analysis

This section describes the experimental environment and results of our analysis. Here, we compare the proposed KGMC with existing rating prediction models and seek to answer the following research questions:

RQ1 Can the performance of the GNN-based rating prediction be improved by utilizing keyword-sharing relationships?
RQ2 How does the performance change in conjunction with the edge addition methods?
RQ3 Does the proposed method effectively handle sparse graphs?
RQ4 How does rating prediction performance change depending on the keyword extraction method?
RQ5 How does the performance of the model depend on the type of GNN layer?
RQ6 Is the keyword utilization method for graphs effective in transfer learning?
RQ7 Does the proposed method provide explainable results that humans can understand?

4.1. Dataset

Experiments were conducted using datasets from eight different domains to examine whether the proposed method works across diverse areas. Data collected from three platforms were used. Among the Amazon review datasets, 5-core datasets (i.e., datasets in which all nodes have 5 or more neighbors) from six domains were used: Movie, Music, Game, Sport, Office, and Grocery. Yelp is a review dataset for various businesses, and 5-core users/items were selected. Finally, Epinions is a review dataset for products written on an online platform and 3-core users/items were selected. Datasets of various sizes, ranging from 25,000 to 1.14 million, were selected. All datasets contained the user ID, item ID, review text, rating, and timestamp. The basic statistics for the datasets are summarized in Table 1. Datasets of various densities were selected to examine the effect of sparsity. Some reviews were excessively long; therefore, the maximum length was limited to 300 words per review to eliminate the influence of outliers. Reviews with more than 300 words were only 3.53% on average across the datasets.

4.2. Baselines

To compare the proposed method with various existing rating prediction methods, three different approaches were selected as baselines: matrix factorization, the review-aware method, and a GNN-based method. Table 2 lists the elements comprising each baseline. A detailed description of each baseline model is provided below.

4.2.1. Matrix factorization

- **NMF** (Lee and Seung, 2000) is a method of predicting unobserved ratings from the user–item rating matrix through non-negative matrix factorization.
- **PMF** (Mnih and Salakhutdinov, 2007) is a probabilistic linear model that transforms matrix factorization into a probabilistic model.
- **SVD++** (Koren, 2008) is a model that extends singular value decomposition based on a rating matrix by using item similarity.

4.2.2. Review-aware models

- **DeepCoNN** (Zheng et al., 2017) is a CNN-based review-aware model that generates user and item representations from review documents and predicts ratings between users and items.
- **NARRE** (Chen et al., 2018) is a model that improves rating prediction performance and identifies useful reviews by applying an attention mechanism to a review-aware model using CNN.

4.2.3. Graph-based models

- **SimGCL** (Yu et al., 2022) is a contrastive learning-based GNN method, which replaces graph augmentations with uniform noise added to the embedding space, resulting in improved recommendation accuracy and training efficiency.
- **XSimGCL** (Yu et al., 2023) is an extremely simplified variant of SimGCL that uses only a path of GNN layers and contrasts the first and the last node embeddings.
- **RGCL** (Shuai et al., 2022) is a GNN-based model that utilizes a review vector extracted using BERT. RGCL utilizes the review embeddings obtained from BERT as edge features.
- **MEGCF** (Liu et al., 2022) integrates semantic-rich entities from multimodal data into a user–item interaction graph and employs a GNN to capture semantic correlations and collaborative filtering signals. Sentiment information from reviews is also used to weight neighbor aggregation, enhancing the effectiveness of MEGCF in modeling multimodal user preferences. However, in our experiment, we only had access to user reviews; therefore, only review text and ratings were used for this model, which is also true for the other baseline and proposed models.
- **AHOR** (Wang et al., 2023) is a GNN-based model that utilizes aspects extracted from review text. Among the graph-based rating prediction models, AHOR is the most recent model that uses reviews.
- **IGMC** (Zhang and Chen, 2020) is a subgraph-based inductive model that captures graph structure information. It is robust for sparse data and transfer learning has been confirmed possible through experiments (Zhang and Chen, 2020).

4.3. Experimental settings

4.3.1. Keyword extraction

Three keyword-extraction methods were used to examine the performance differences associated with the keyword-extraction methods: KeyBERT, TF-IDF, and TextRank. Each document was created by combining the reviews of each user and item, followed by keyword extraction. A maximum of five keywords were extracted from each document, and only single nouns and adjectives were used as keyword candidates. KeyBERT used the *distilbert-base-nli-mean-tokens* pre-trained model (Reimers and Gurevych, 2019) from the Sentence Transformers library.[2] The extracted keywords were assigned to each user and item node and were used to build the keyword-enhanced subgraphs.

[2] https://www.sbert.net/.

4.3.2. Evaluation metric

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{5}$$

The root mean square error (RMSE), as described in Eq. (5), was used as the evaluation metric. y_i and $\hat{y}_i$ denote the actual and predicted ratings of the user–item interaction i, respectively, and N denotes the total number of interactions in the test dataset. In our rating prediction task, a lower RMSE indicates a better performance.

$$NDCG@k = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{log_2(p_i+1)} \tag{6}$$

$$MRR@k = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{p_i+1} \tag{7}$$

We also evaluated our model with normalized discounted cumulative gain (NDCG) and mean reciprocal rank (MRR), considering that recommender systems are commonly evaluated using top-k metrics. Both metrics have been widely used in recommendation system studies (Wu et al., 2022b). When applying a recommendation system to a web application, multiple item candidates are shown to the user, and only some of them are selected. Therefore, in order to evaluate in a similar way to the real-world web environment, prior studies randomly generated negative pairs to apply top-k metrics (Wang et al., 2022; Zhang et al., 2023; Yang et al., 2022). In our datasets, as there were no negative pairs, we also had to randomly generate them. Therefore, for the NDCG and MRR evaluations, we selected two thousand random users and their last positive interactions, and generated 99 randomly sampled negative interactions, which do not exist in the test datasets, as it was done identically in a prior recommender system study (Wang et al., 2022). Eqs. (6) and (7) present how to evaluate with NDCG@k and MRR@k. N is the total number of test users and p_i is the position of a positive item in the $top-k$ list of the ith user.

4.3.3. Train setting

Our KGMC model was implemented using PyTorch (Paszke et al., 2019) and DGL (Wang et al., 2019). The datasets were split into train (60%), validation (20%), and test (20%) subsets based on chronological order. The mean squared error (MSE) was used for the loss function, as our evaluation metric was RMSE. Adam optimizer (Kingma and Ba, 2015) was used with initial learning rates of [2e−3, 1e−3, 5e−3]. The batch size was fixed at 128, and the node-embedding dimensions in the hidden layers of the GNN were fixed at 32; 1-hop enclosing subgraphs were used for all of the datasets. After subgraph extraction, we randomly dropped out the edges of each subgraph with a probability of 0.2 during training. Four R-GCN layers were used in the study. These hyperparameters were also used in the IGMC baseline experiments for a fair comparison. Each experiment was repeated five times and the mean was computed for reporting.

4.4. Experiment results

4.4.1. Performance comparison to baselines

To answer **RQ1**, we compared the performance of KGMC with the performance of the baseline models. Table 3 shows the experimental results comparing the baseline models and KGMC. The numbers shown in bold indicate the best model outcome in each domain, and the underlined numbers indicate the best baseline outcomes. Among the three variants of KGMC, created by varying the edge types, KGMC with the homogeneous type edges (KGMC-homo), produced better performance than KGMC with the heterogeneous type edges (KGMC-hetero) and KGMC with the mixed type edges (KGMC-mixed). Furthermore, KGMC-homo achieved the best-performing results, outperforming each of the baseline models across the eight datasets, without exception. Among the three keyword extraction techniques, all three variants of KGMC were implemented via KeyBERT for the results reported in the table.

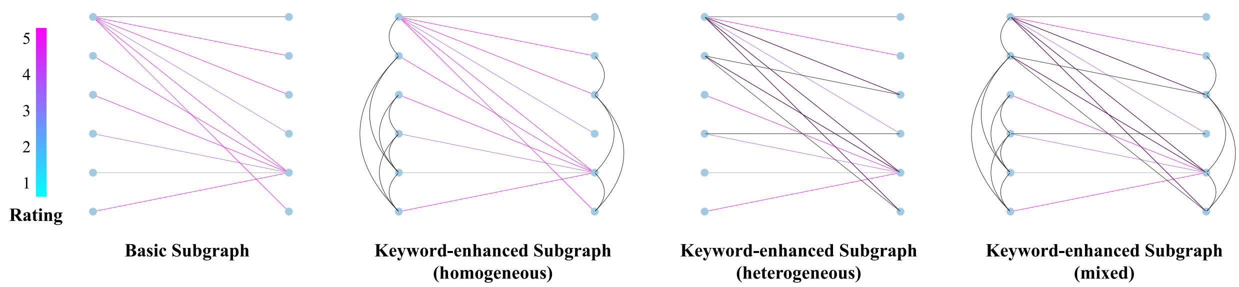

Fig. 8. Visualization of keyword-enhanced subgraphs based on alternative edge addition methods.

Table 3
RMSE performance of the baseline models and KGMC.

Model\Dataset	Movie	Music	Game	Sport	Office	Grocery	Yelp	Epinions
NMF	1.1050	0.7846	1.2043	1.1653	1.1214	1.1608	1.3084	1.2284
SVD++	1.0250	0.6805	1.1139	1.0033	0.9926	1.0789	1.1965	1.1913
PMF	1.1211	0.6940	1.1491	1.0211	1.0140	1.1361	1.3422	1.2212
DeepCoNN	0.9272	0.6160	1.0472	0.9675	0.9737	1.1012	1.2101	1.1890
NARRE	0.8919	0.5906	1.0387	0.9932	0.9865	1.0954	1.2959	1.1926
SimGCL	0.9689	0.6589	1.0904	0.9836	0.9722	1.1301	1.2100	1.1920
XSimGCL	0.9899	0.6782	1.1033	0.9923	0.9836	1.1126	1.2366	1.2420
RGCL	0.9998	0.6700	1.1140	0.9978	0.9905	1.0756	1.1742	1.1880
MEGCF	0.9625	0.6201	1.0870	0.9609	0.9110	1.1082	1.2110	1.1713
AHOR	0.9718	0.6598	1.0867	0.9784	0.9719	1.1096	1.1905	1.1929
IGMC	<u>0.8659</u>	<u>0.5828</u>	<u>0.9990</u>	<u>0.9210</u>	<u>0.8903</u>	<u>1.0192</u>	<u>1.0661</u>	<u>1.1184</u>
KGMC-hetero	0.8374	0.5499	0.9826	0.9011	0.8698	1.0032	1.0531	1.1098
KGMC-homo	**0.8331**	**0.5487**	**0.9804**	**0.9003**	**0.8692**	**1.0013**	**1.0482**	**1.1080**
KGMC-mixed	0.8357	0.5508	0.9823	0.9012	0.8707	1.0025	1.0495	1.1109
Improvement	3.78%	5.85%	1.85%	2.25%	2.37%	1.76%	1.69%	0.85%

The best results are highlighted in bold, and the best results among the baselines are underlined. Each experiment was repeated five times and the *mean* was reported. The improvement shows the percentage of improvement made by the best model over the second-best model for each dataset. T-tests confirmed that the improvements were all significant at a p-value < 0.01.

On average, KGMC achieved performance improvement by an average of 2.6% when compared with the best baseline model. For each of the datasets, its performance was better than any of the baseline models, without exception. Out of all of the datasets, KGMC showed a performance improvement by 12.4% compared with the best MF baseline model, 7.6% compared with the best review-aware baseline model, and 2.6% compared with the best graph-based baseline model, on average. In a review-based rating prediction study, a relative performance improvement above 1% is regarded as a significant change (Tay et al., 2018; Li et al., 2021; Shuai et al., 2022). In our study, our proposed model achieved more than 1% performance improvement in every domain, except Epinions (0.85%), over IGMC which achieved the second-best performance in every domain dataset, as shown in Table 3.

Regarding the experimental results of the baseline models, IGMC achieved the highest performance for every dataset. Thus, we have confirmed that the inductive graph-based approach is superior for rating prediction. The performance improvement made by IGMC over the other baseline models was the largest in the Music dataset and smallest in the Epinions dataset. Epinions is characterized as a 3-core dataset, which means that all nodes have at least 3 neighbors. The inductive graph method appears to be more effective when k is large in the k-core settings. From this observation, we attribute the relatively smaller performance gains of the KGMC models in Epinions to the nature of the dataset.

4.4.2. Comparison of the edge addition methods

To answer **RQ2**, three KGMC variants were created and tested using alternative edge generation methods. As shown in Table 3, in all of the cases that used additional keyword co-occurrence edges, higher performance was achieved over IGMC, which used rating edges only. This proves that the keyword information we added helps accurately predict ratings. The highest performance was achieved only when homogeneous edges were added across all of the datasets. From these results, we can infer that a learning method that includes the relationships between homogeneous nodes is beneficial in enhancing the inductive graph-based method.

We visualized samples of keyword-enhanced subgraphs for intuitive understanding. Fig. 8 shows sample subgraphs extracted from the Yelp dataset. We can observe that the subgraph structure changes depending on how keyword co-occurrence edges are added, and we found that adding homogeneous edges improves performance because it enables our model to additionally utilize relationships between homogeneous nodes that are not found in conventional rating relationships. On the other hand, adding heterogeneous edges has the potential to perturb the explicit rating relationship, so the performance improvement was limited.

4.4.3. Top-K performance evaluation

To further investigate the performance superiority of KGMC relative to IGMC, we compared the two models by applying the top-k evaluation metrics. Table 4 shows the experimental results between IGMC and KGMC with top-k metrics. KGMC outperformed IGMC in all evaluation metrics, leading us to conclude that more appropriate items will be recommended when KGMC is applied to commercial recommendation systems, where top-K choices are likely to play a critical role for their success.

4.4.4. Data sparsity

To answer **RQ3**, we compared performance improvements based on the density of the datasets. Even on very sparse datasets like Sport and Office, KGMC showed significant performance gains. IGMC has

Table 4
Top-k performance comparison of IGMC and KGMC.

Model	Metric	Music	Games	Office	Sports	movie	Grocery	Epinions	Yelp
IGMC	NDGC@5	0.1115	0.1705	0.1490	0.0977	0.1556	0.0969	0.0285	0.0871
	NDGC@10	0.1237	0.2011	0.1487	0.1298	0.1775	0.1141	0.0561	0.1098
	NDGC@20	0.1403	0.2276	0.1936	0.1508	0.2077	0.1357	0.0647	0.1316
	MRR@5	0.1023	0.1433	0.1300	0.0873	0.1321	0.0863	0.0232	0.0758
	MRR@10	0.1073	0.1563	0.1383	0.0998	0.1410	0.0932	0.0307	0.0844
	MRR@20	0.1120	0.1635	0.1449	0.1068	0.1491	0.0993	0.0353	0.0907
KGMC	NDGC@5	**0.1224**	**0.1933**	**0.1600**	**0.1174**	**0.2124**	**0.1159**	**0.0629**	**0.1015**
	NDGC@10	**0.1470**	**0.2148**	**0.1795**	**0.1465**	**0.2386**	**0.1288**	**0.0770**	**0.1174**
	NDGC@20	**0.1705**	**0.2391**	**0.2171**	**0.1688**	**0.2659**	**0.1436**	**0.1003**	**0.1584**
	MRR@5	**0.1061**	**0.1699**	**0.1381**	**0.1070**	**0.1925**	**0.1080**	**0.0522**	**0.0907**
	MRR@10	**0.1163**	**0.1786**	**0.1499**	**0.1146**	**0.2034**	**0.1133**	**0.0579**	**0.0992**
	MRR@20	**0.1229**	**0.1851**	**0.1576**	**0.1232**	**0.2106**	**0.1175**	**0.0642**	**0.1089**

The best results are highlighted in bold.

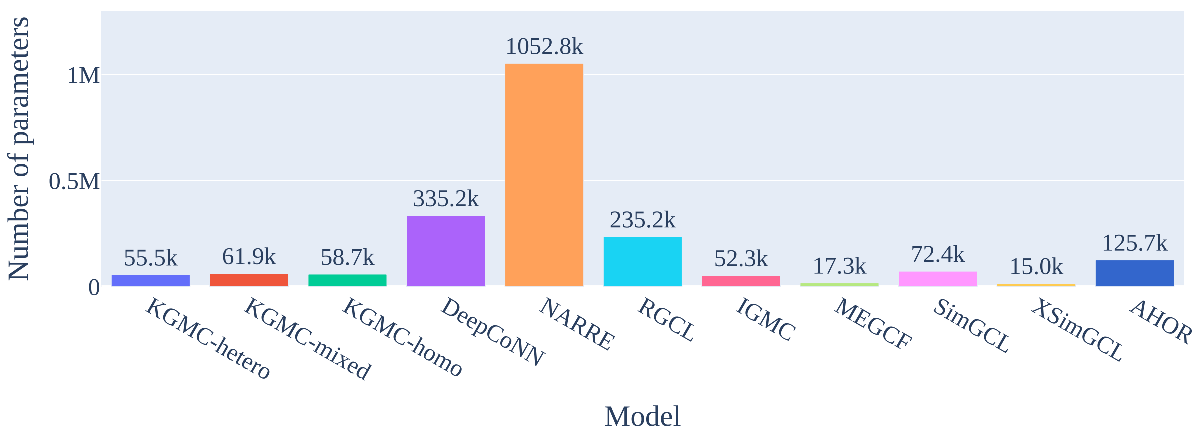

Fig. 9. Comparing the number of parameters between models.

been shown to be effective for sparse datasets in a prior study (Zhang and Chen, 2020). KGMC outperformed IGMC in the datasets of various densities in our study, strongly supporting the robustness of KGMC against sparse datasets.

4.5. Comparing the number of learnable parameters

The number of model parameters is an important factor in determining training time and memory usage. We compared the number of parameters in the baseline and our proposed models to analyze not only the performance but also the cost of the models. Fig. 9 shows the number of parameters for the models we employed in our experiments. The analysis confirms that KGMC is not only superior in performance but also efficient in terms of the number of parameters. Review-based models such as RGCL, NARRE, and DeepCoNN have a very large number of parameters, which require more computational resources for training and inference. MEGCF, which is based on LightGCN, has relatively fewer parameters, but is inferior to KGMC in terms of performance.

4.6. Ablation study

We performed ablation studies to evaluate the effectiveness of our method. Two ablation studies were conducted from two perspectives: keyword extraction method and GNN layer type.

4.6.1. Comparison of keyword-extraction methods

To answer **RQ4**, which questions if the rating prediction performance changes depending on the keyword extraction method, the performance of KGMC was compared using the three keyword-extraction methods: KeyBERT, TextRank, and TF-IDF. The experiment was performed only for the homogeneous type of edge addition as it was found the most effective. Table 5 presents the experimental results of KGMC obtained while varying the keyword-extraction methods. The keyword extraction method that achieved the highest performance across the datasets was inconsistent. There was no single keyword extraction method that performed the best all the time and the performance differences were marginal. KeyBERT achieved the highest performance on the five datasets, indicating that it can perform as well or better than TF-IDF, except for the Office dataset, without referring to all documents. The pre-trained model was already trained with many other documents and was able to extract high-quality keywords.

The differences in the datasets resulted in different appearance patterns of meaningful keywords in each domain. To compare keyword qualities of the keyword-extraction methods, we extracted and compared the top 10 most frequently used keywords in each domain. Tables 6 and 7 compare the keywords of users and items in different domains. Keywords that express users' tastes or item characteristics are marked in bold. Assessing the qualities of keywords in a specific domain is a challenging task (Abulaish et al., 2022). We manually marked domain-specific keywords and excluded expressions that indicated general positive preferences such as *good*, *great* and *love*. A relatively clear difference in keyword quality was observed in the Grocery and Yelp datasets where KeyBERT exhibited the highest performance. Compared with other keyword-extraction methods, domain-specific keywords were extracted at a high rate from keywords extracted with KeyBERT. In the case of the Yelp and Grocery datasets, where KeyBERT achieved the highest performance, more domain-specific keywords were observed with KeyBERT than with the other keyword-extraction methods. The item keywords extracted from the Grocery dataset using TF-IDF had a high proportion of domain-specific keywords, but there were few domain-specific keywords among the user keywords. However, in the case of Office and Movie, there

Table 5
RMSE performance of KGMC using the keyword-extraction methods.

Method\Dataset	Movie	Music	Game	Sport	Office	Grocery	Yelp	Epinions
KeyBERT	0.8331	0.5487	**0.9804**	**0.9003**	0.8692	**1.0013**	**1.0482**	**1.1080**
TextRank	0.8332	**0.5476**	0.9810	0.9013	**0.8678**	1.0074	1.0493	1.1496
TF-IDF	**0.8327**	0.5488	0.9827	0.9022	0.8696	1.0058	1.0505	1.1511

The best results are highlighted in bold.

Table 6
Keyword samples from Yelp and Grocery datasets.

Method	KeyBERT		TextRank		TFIDF	
	User	Item	User	Item	User	Item
Grocery	flavor	**tea**	great	great	great	**coffee**
	chocolate	flavor	taste	good	good	**tea**
	tea	**coffee**	good	taste	**coffee**	**chocolate**
	coffee	**chocolate**	flavor	flavor	**tea**	**candy**
	cookies	taste	like	like	love	**sauce**
	taste	love	**coffee**	love	product	**sugar**
	favorite	delicious	**tea**	**tea**	taste	**salt**
	vanilla	favorite	product	**coffee**	delicious	good
	delicious	**cookies**	love	product	flavor	**bars**
	cinnamon	**vanilla**	**chocolate**	best	like	**organic**
Yelp	**pizza**	**burger**	great	great	good	food
	burger	**pizza**	food	food	great	**pizza**
	steak	**salad**	good	good	food	**chicken**
	dinner	**steak**	service	service	place	**coffee**
	salad	**dinner**	like	order	**pizza**	**car**
	sushi	lunch	place	like	delicious	**breakfast**
	taco	**sandwich**	order	place	**husband**	store
	restaurant	**bacon**	**chicken**	**chicken**	really	**sushi**
	sandwich	**taco**	time	time	**wife**	good
	shrimp	restaurant	best	**pizza**	order	**burger**

Table 7
Keyword samples from Office and Movie datasets.

Method	KeyBERT		TextRank		TFIDF	
	User	Item	User	Item	User	Item
Office	love	love	great	great	great	great
	great	great	good	good	good	good
	good	happy	work	work	product	**pen**
	printer	good	**pen**	**pen**	love	**ink**
	pencil	**quality**	**quality**	product	work	**printer**
	ink	nice	nice	work	**pen**	product
	nice	printer	product	**quality**	**price**	work
	perfect	amazon	love	nice	nice	love
	happy	favorite	**color**	**ink**	**quality**	**paper**
	favorite	**ink**	**ink**	**color**	perfect	**card**
Movie	movie	movie	movie	movie	movie	movie
	love	love	great	great	great	**series**
	great	favorite	good	good	good	great
	good	**dvd**	film	love	love	good
	favorite	great	love	film	film	love
	dvd	fun	story	story	excellent	fun
	fun	happy	like	**series**	**series**	film
	best	**christmas**	films	like	ok	**season**
	christmas	good	**series**	excellent	**classic**	**christmas**
	enjoy	**husband**	**classic**	action	awesome	**classic**

was no significant difference between the sample keyword sets. From these results, we can infer that domain-specific keywords affect the performance of rating prediction. Also, it should be noted that there are limitations to evaluating the quality of the entire keyword set using sample keywords; however, significant insights can be observed through the keyword sampling analysis.

4.6.2. Comparison of GNN layers

To answer **RQ5**, we experimented with utilizing the keyword-enhanced graph of KGMC with different GNN layers to observe performance variations across GNN layers. We experimented with five different GNN layer types, including R-GCN used in KGMC. LightGCN (He et al., 2020) is a lightweight GCN model, GIN (Xu et al., 2019) is a model utilizing structural information based on graph isomorphism, GATv2 (Brody et al., 2022) is an improvement of the graph attention network to learn the importance of neighbors, and HGT (Hu et al., 2020) is a model to reflect the characteristics of heterogeneous graphs. The experiment was performed only for the homogeneous type of edge addition.

Table 8 shows the RMSE performance of each variant of GNN layers as well as the number of learnable parameters. HGT achieved the highest performance on five datasets, while R-GCN achieved the highest performance on three datasets. However, the performance gap between the two GNNs is marginal on all datasets, and HGT has twice as many as R-GCN in terms of the number of learnable parameters. These results show that our proposed model utilizing R-GCN is efficient from a computational cost perspective.

4.7. Cross-domain experiment

To verify how well KGMC maintains the capability of generalizability and transferability, which are known as the strengths of IGMC (Zhang and Chen, 2020), cross-domain experiments were conducted. As shown in Tables 6 and 7, each domain contains different keywords, and in a general approach, these differences in keywords make transfer learning difficult. However, we conducted experiments to check whether KGMC can overcome these domain-specific keyword differences and perform cross-domain transfer learning. Two experiments were conducted: direct-transferring, which performs rating prediction without fine-tuning, and fine-tuning, which trains the model with a target dataset again (Zhu et al., 2021). The pre-trained models were trained on three datasets (i.e., Yelp, Office, Grocery) and the cross-domain experiments were performed on the other five datasets (i.e., Music, Game, Sport, Movie, Epinions).

We tested the cross-domain transferability and generalization ability of KGMC to answer **RQ6**, which asks if the keyword utilization method for graphs is effective in transfer learning. Table 9 presents the performance of direct-transferring and Table 10 presents the performance of fine-tuning. We tested both models in the same way to compare the cross-domain performances of IGMC and KGMC.

In the direct-transferring experiment, the pre-trained models were evaluated with a different domain dataset without any fine-tuning. For example, the IGMC-Yelp model was created by training IGMC using the Yelp dataset and tested using a different dataset while the IGMC-original model was created by training IGMC using the domain dataset to which it was applied for testing. The results of the direct-transferring experiment show that both IGMC and KGMC can perform well similar to the original training results. Overall, KGMC showed higher performance than IGMC for all of the datasets in the direct-transferring test, as shown in Table 9. Remarkably, the KGMC direct-transferring performance, in some cases, exceeded that of the IGMC-original, indicating that KGMC inherits the generalization ability of IGMC and achieves higher performance by utilizing additional information.

As presented in Table 10, the fine-tuning experimental results show that KGMC achieved higher performance than IGMC for all of the datasets. In most cases, pre-training with the Grocery dataset achieved the highest performance, probably because the Grocery dataset was the largest dataset and contained diverse patterns. However, for the

Table 8
RMSE performance of KGMC with alternative GNN layers.

GNN	#parameters	Movie	Music	Game	Sport	Office	Grocery	Yelp	Epinions
R-GCN	58.8K	<u>0.8331</u>	<u>0.5487</u>	**0.9804**	**0.9003**	<u>0.8692</u>	<u>1.0013</u>	<u>1.0482</u>	**1.1080**
LightGCN	33.2K	1.0434	0.6843	1.1399	1.0125	1.0060	1.1256	1.2443	1.2607
GIN	33.2K	1.2122	0.7324	1.3078	1.0361	1.1216	1.2899	1.5773	1.2627
GATv2	46.6K	1.0375	0.6815	1.1384	1.0121	1.0048	1.1246	1.2402	1.2575
HGT	116.7K	**0.8231**	**0.5448**	<u>0.9843</u>	<u>0.9012</u>	**0.8607**	**0.9975**	**1.0446**	<u>1.1893</u>

The best results are highlighted in bold, and the second-best results are underlined.

Table 9
RMSE performance of the direct-transferring.

Model\Dataset	Music	Game	Sport	Movie	Epinions
IGMC-original	0.5848	1.0044	0.9240	0.8659	1.1184
IGMC-Yelp	0.6557	1.0270	0.9559	0.8928	1.1264
IGMC-Office	0.5904	1.0056	0.9230	0.8588	1.1642
IGMC-Grocery	0.6107	1.0026	0.9248	0.8647	1.1562
KGMC-original	**0.5487**	**0.9804**	**0.9003**	0.8331	**1.1080**
KGMC-Yelp	0.6196	0.9989	0.9368	0.8615	<u>1.1168</u>
KGMC-Office	<u>0.5692</u>	0.9861	0.9046	<u>0.8317</u>	1.1316
KGMC-Grocery	0.5710	<u>0.9827</u>	<u>0.9026</u>	**0.8286**	1.1177
Improvement	3.59%	1.98%	2.22%	3.52%	0.86%

The best results are highlighted in bold, and the second-best results are underlined. The improvement shows the percentage of improvement made by KGMC over IGMC for each dataset.

Table 10
RMSE performance of the fine-tuning.

Model\Dataset	Music	Game	Sport	Movie	Epinions
IGMC-original	0.5828	0.9990	0.9210	0.8659	1.1184
IGMC-Yelp	0.5805	0.9950	0.9200	0.8558	1.1171
IGMC-Office	0.5811	0.9954	0.9200	0.8562	1.1181
IGMC-Grocery	0.5797	0.9955	0.9195	0.8583	1.1176
KGMC-original	0.5487	0.9804	0.9003	0.8331	**1.1080**
KGMC-Yelp	0.5464	0.9797	**0.8986**	0.8236	<u>1.1081</u>
KGMC-Office	<u>0.5456</u>	<u>0.9784</u>	0.8997	<u>0.8226</u>	1.1121
KGMC-Grocery	**0.5440**	**0.9776**	<u>0.8990</u>	**0.8176**	1.1092
Improvement	6.15%	1.75%	2.28%	4.46%	0.76%

The best results are highlighted in bold, and the second-best results are underlined. Each experiment was repeated five times and the *mean* was reported. The improvement shows the percentage of improvement made by KGMC over IGMC for each dataset. T-tests confirmed that the improvements were all significant at a p-value < 0.01.

Epinions dataset, there was no performance improvement through fine-tuning probably due to the structural difference in the k-core setting. Overall, the results indicate that adding additional information in the form of graph edges is also effective for fine-tuning except for the datasets with very small k-core settings. The datasets used for pre-training were 5-core, which were different from the Epinions dataset in terms of graph structure. Therefore, we attributed the lack of performance improvement with transfer learning on Epinions, a 3-core dataset, to the fact that transfer learning in GNNs is highly correlated with graph structure information (Zhu et al., 2021). The proposed method can be similarly applied not only to keywords but also to cases where features are shared between nodes. Therefore, the proposed edge-enhancing method can be extended to various types of data.

4.8. Keyword-based explainability

Another advantage of KGMC is its explainability. Along with high predictive power, the ability to interpret results is crucial for recommender systems. Explainable recommendation models with knowledge graphs have previously been proposed (Li et al., 2022). Going further, KGMC not only improves performance by using the extracted keywords but also identifies those keywords that connect users and items through the keywords included in the subgraph. To answer **RQ7**, which asks if the proposed method is explainable, we demonstrate keyword-based explainability using several examples. For this purpose, we introduce a method for explaining results by comparing keywords of target users and items with keywords of neighboring nodes in the keyword-enhanced subgraph.

Table 11 presents a list of keywords included in the keyword-enhanced subgraphs extracted from the four datasets. The table contains the target user keywords, item keywords, neighboring keywords, and the appearance frequencies of users and items from each sample subgraph. The five most frequently used keywords among the neighboring node keywords have been selected as neighboring keywords. Those keywords that occur commonly in the target user, item, and neighbor are displayed in bold. From the table on the left, we can see that the keyword *beef* appears in common in the subgraph extracted from the Yelp dataset. The keywords *beef* and *sandwich* frequently appear among the keywords of the neighboring nodes. Based on these results, we can infer that the *beef* keyword connects the user and the item and it is an important component of the subgraph. From the table on the right, we can see that the keyword *chicken* appears in common in the case of samples from the Grocery dataset. Among the keywords of the neighboring nodes, the keywords *chicken, soup* appear frequently. Thus, it can be inferred that the subgraph is about chicken soup. As shown by these examples, KGMC has the advantage of enabling an intuitive interpretation of results without additional processes.

As demonstrated earlier, keyword-based methods are useful to intuitively understand the results. More specifically, these results can be used from two perspectives. The first is to understand the detailed tastes of users and the characteristics of items. The tastes of users in e-commerce platforms are becoming increasingly diverse, and it is important to understand their tastes in detail. The detailed tastes of users can be understood through the keyword-based interpretations we illustrated. The second is to use the insights gained from the keywords to develop new items. It is an important and risky task to develop new products that can satisfy users. By capturing keyword pairs that appear together within the subgraph, they can be used to develop insights into new products that can better appeal to users.

5. Conclusion

In this study, we proposed KGMC, which is a new keyword-enhanced graph-based rating prediction model. We created user and item documents based on reviews, and extracted keywords using KeyBERT, TFIDF, and TextRank, separately. The extracted keywords were used to create additional edges in the subgraph consisting of the target user, item, and its neighbors. In this way, we sought to utilize the latent information contained in the review while maintaining its inductive characteristics.

In our experiment, we found that the proposed model was superior to the state-of-the-art baselines, including the most recent graph-based models such as RGCL and MEGCF. In addition, cross-domain experiments confirmed that generalizability and transferability, which are the strengths of the inductive model, were effectively maintained. In our ablation studies, we observed that the proposed model performed the best when utilizing keyword co-occurrence edges between homogeneous nodes. In addition, we compared the prediction performances while varying the keyword extraction methods. For most datasets, KeyBERT showed the best performance.

Table 11
Keywords in subgraph samples.

Sample	Yelp	User ID 4602	Item ID 1835			Grocery	User ID 103161	Item ID 9760		
Category	User keywords	Item keywords	Neighboring keywords	Users count	Items count	User keywords	Item keywords	Neighboring keywords	Users count	Items count
	mediocre	mushroom	burger	3	4	flavor	onions	**chicken**	31	4
	beef	**beef**	**beef**	7	1	**chicken**	**chicken**	soup	15	7
	chef	onions	salad	3	1	chili	noodles	flavor	6	2
	pasta	sandwich	sandwich	7	3	popcorn	soup	tasty	4	4
	lunch	steak	pizza	3	2	tasty	macaroni	chili	2	3
Sample	Game	User ID 13610	Item ID 1488			Sport	User ID 33780	Item ID 5627		
Category	User keywords	Item keywords	Neighboring keywords	Users count	Items count	User keywords	Item keywords	Neighboring keywords	Users count	Items count
	irritating	toadstool	**nintendo**	7	6	wide	secure	**bike**	12	4
	gameboy	**nintendo**	**wii**	7	7	long	**tire**	**tire**	6	3
	mario	**gameboy**	gamecube	5	3	fat	**bike**	cheap	4	2
	nintendo	**wii**	**gameboy**	5	3	**tire**	efficient	easy	2	4
	wii	weakest	mario	4	8	**bike**	cheap	efficient	2	3

KGMC can provide an intuitive explanation using the keywords included in the subgraphs. This method not only makes it possible to understand the various tastes of users, but also provides insights through the keyword pairs that appear together within the subgraph for new product development. Furthermore, it is worth noting that the proposed method can also mitigate the data sparsity problem as latent information between additional users and items extracted from side information is actively exploited during the learning process (Ahmadian et al., 2022; Chen et al., 2022; Ali et al., 2023). Indeed, developing robust models on the sparse dataset is one of the main research topics in recommender systems, and the proposed model shows promising results for tapping into the potential of essential aspects of the inductive graph-based learning approach for the enhancement of recommendation accuracy even on a sparse dataset.

For future research, we are in the process of designing a method to identify and evaluate keywords with high importance for each domain and reflect the varying importance of each keyword in the model. This keyword importance identification method is expected to further enhance the explainability of the proposed model.

CRediT authorship contribution statement

Donghee Han: Conceptualization, Methodology design, Data curation, Experiments. **Daehee Kim:** Data curation, Experiments. **Keejun Han:** Supervision, Writing – review & editing, Resources. **Mun Yong Yi:** Supervision, Writing – review & editing, Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by Korea Institute of Planning and Evaluation for Technology in Food, Agriculture and Forestry (IPET) and Korea Smart Farm R&D Foundation (KosFarm) through Smart Farm Innovation Technology Development Program, funded by Ministry of Agriculture, Food and Rural Affairs (MAFRA) and Ministry of Science and ICT (MSIT), Rural Development Administration (RDA) (421043-04)

References

Abulaish, M., Fazil, M., Zaki, M.J., 2022. Domain-specific keyword extraction using joint modeling of local and global contextual semantics. ACM Trans. Knowl. Discov. Data (2022) 16 (4), URL https://doi.org/10.1145/3494560.

Afoudi, Y., Lazaar, M., Hmaidi, S., 2023. An enhanced recommender system based on heterogeneous graph link prediction. Eng. Appl. Artif. Intell. 124, 106553, URL https://www.sciencedirect.com/science/article/pii/S0952197623007376.

Ahmadian, S., Ahmadian, M., Jalili, M., 2022. A deep learning based trust- and tag-aware recommender system. Neurocomputing 488, 557–571, URL https://www.sciencedirect.com/science/article/pii/S0925231221017410.

Ali, W., Kumar, J., Mawuli, C.B., She, L., Shao, J., 2023. Dynamic context management in context-aware recommender systems. Comput. Electr. Eng. 107, 108622, URL https://www.sciencedirect.com/science/article/pii/S0045790623000472.

Bao, Y., Fang, H., Zhang, J., 2014. Topicmf: Simultaneously exploiting ratings and reviews for recommendation. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence. AAAI '14, AAAI Press, pp. 2–8, URL https://ojs.aaai.org/index.php/AAAI/article/view/8715.

Bing, R., Yuan, G., Zhu, M., Meng, F., Ma, H., Qiao, S., 2023. Heterogeneous graph neural networks analysis: a survey of techniques, evaluations and applications. Artif. Intell. Rev. 56 (8), 8003–8042, URL https://doi.org/10.1007/s10462-022-10375-2.

Brody, S., Alon, U., Yahav, E., 2022. How attentive are graph attention networks? In: International Conference on Learning Representations. URL https://openreview.net/forum?id=F72ximsx7C1.

Chen, M., Huang, C., Xia, L., Wei, W., Xu, Y., Luo, R., 2023. Heterogeneous graph contrastive learning for recommendation. In: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining. WSDM '23, Association for Computing Machinery, New York, NY, USA, pp. 544–552, URL https://doi.org/10.1145/3539597.3570484.

Chen, Y., Qiu, X., Ma, C., Xu, Y., Sun, Y., 2022. A recommender system fused with implicit social information through network representation learning. Comput. Electr. Eng. 100, 107897, URL https://www.sciencedirect.com/science/article/pii/S0045790622001835.

Chen, C., Zhang, M., Liu, Y., Ma, S., 2018. Neural attentional rating regression with review-level explanations. In: Proceedings of the 2018 World Wide Web Conference. WWW '18, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, pp. 1583–1592, URL https://doi.org/10.1145/3178876.3186070.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186, URL https://aclanthology.org/N19-1423.

Gajbhiye, G.O., Nandedkar, A.V., 2022. Generating the captions for remote sensing images: A spatial-channel attention based memory-guided transformer approach. Eng. Appl. Artif. Intell. 114, 105076, URL https://www.sciencedirect.com/science/article/pii/S0952197622002317.

Gao, J., Lin, Y., Wang, Y., Wang, X., Yang, Z., He, Y., Chu, X., 2020. Set-sequence-graph: A multi-view approach towards exploiting reviews for recommendation. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. CIKM '20, Association for Computing Machinery, New York, NY, USA, pp. 395–404, URL https://doi.org/10.1145/3340531.3411939.

Ge, Y., Zhao, S., Zhou, H., Pei, C., Sun, F., Ou, W., Zhang, Y., 2020. Understanding echo chambers in E-commerce recommender systems. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '20, Association for Computing Machinery, New York, NY, USA, pp. 2261–2270, URL https://doi.org/10.1145/3397271.3401431.

Gleich, D.F., 2015. PageRank beyond the web. SIAM Rev. 57 (3), 321–363, URL https://doi.org/10.1137/140976649.

Grootendorst, M., 2020. Keybert: Minimal keyword extraction with bert.. URL https://doi.org/10.5281/zenodo.4461265.

Hamilton, W.L., Ying, R., Leskovec, J., 2017. Inductive representation learning on large graphs. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS '17, Curran Associates Inc., Red Hook, NY, USA, pp. 1025–1035, URL https://proceedings.neurips.cc/paper_files/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf.

Han, X., Huang, Z., An, B., Bai, J., 2021. Adaptive transfer learning on graph neural networks. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. KDD '21, Association for Computing Machinery, New York, NY, USA, pp. 565–574, URL https://doi.org/10.1145/3447548.3467450.

He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M., 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '20, Association for Computing Machinery, New York, NY, USA, pp. 639–648, URL https://doi.org/10.1145/3397271.3401063.

Hu, Z., Dong, Y., Wang, K., Sun, Y., 2020. Heterogeneous graph transformer. In: Proceedings of the Web Conference 2020. WWW '20, Association for Computing Machinery, New York, NY, USA, pp. 2704–2710, URL https://doi.org/10.1145/3366423.3380027.

Jiang, R., Chiappa, S., Lattimore, T., György, A., Kohli, P., 2019. Degenerate feedback loops in recommender systems. In: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society. AIES '19, Association for Computing Machinery, New York, NY, USA, pp. 383–390, URL https://doi.org/10.1145/3306618.3314288.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization.. In: Bengio, Y., LeCun, Y. (Eds.), International Conference on Learning Representations. URL http://arxiv.org/abs/1412.6980.

Koren, Y., 2008. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '08, Association for Computing Machinery, New York, NY, USA, pp. 426–434, URL https://doi.org/10.1145/1401890.1401944.

Kumar, Z.Y.Z.W.S.Z.A.B., 2022. Integrating reviews and ratings into graph neural networks for rating prediction. J. Ambient Intell. Humaniz. Comput. URL https://doi.org/10.1007/s12652-021-03626-7.

Lee, D., Seung, H.S., 2000. Algorithms for non-negative matrix factorization. In: Leen, T., Dietterich, T., Tresp, V. (Eds.), Advances in Neural Information Processing Systems. vol. 13, MIT Press, URL https://proceedings.neurips.cc/paper/2000/file/f9d1152547c0bde01830b7e8bd60024c-Paper.pdf.

Li, X.-H., Cao, C.C., Shi, Y., Bai, W., Gao, H., Qiu, L., Wang, C., Gao, Y., Zhang, S., Xue, X., Chen, L., 2022. A survey of data-driven and knowledge-aware explainable AI. IEEE Trans. Knowl. Data Eng. 34 (1), 29–49, URL https://ieeexplore.ieee.org/document/9050829.

Li, Z., Cheng, W., Kshetramade, R., Houser, J., Chen, H., Wang, W., 2021. Recommend for a reason: Unlocking the power of unsupervised aspect-sentiment co-extraction. In: Findings of the Association for Computational Linguistics: EMNLP 2021. Association for Computational Linguistics, Punta Cana, Dominican Republic, pp. 763–778, URL https://aclanthology.org/2021.findings-emnlp.66.

Lin, Z., Tian, C., Hou, Y., Zhao, W.X., 2022. Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In: Proceedings of the ACM Web Conference 2022. WWW '22, Association for Computing Machinery, New York, NY, USA, pp. 2320–2329, URL https://doi.org/10.1145/3485447.3512104.

Liu, Y., Ren, Z., Zhang, W.-N., Che, W., Liu, T., Yin, D., 2020. Keywords generation improves E-commerce session-based recommendation. In: Proceedings of the Web Conference 2020. WWW '20, Association for Computing Machinery, New York, NY, USA, pp. 1604–1614, URL https://doi.org/10.1145/3366423.3380232.

Liu, K., Xue, F., Guo, D., Wu, L., Li, S., Hong, R., 2022. MEGCF: Multimodal entity graph collaborative filtering for personalized recommendation. ACM Trans. Inf. Syst. 1–26, URL https://dl.acm.org/doi/10.1145/3544106.

Mihalcea, R., Tarau, P., 2004. TextRank: Bringing order into text. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Barcelona, Spain, pp. 404–411, URL https://aclanthology.org/W04-3252.

Mikolov, T., Chen, K., Corrado, G.S., Dean, J., 2013. Efficient estimation of word representations in vector space. URL http://arxiv.org/abs/1301.3781.

Mnih, A., Salakhutdinov, R.R., 2007. Probabilistic matrix factorization. In: Platt, J., Koller, D., Singer, Y., Roweis, S. (Eds.), Advances in Neural Information Processing Systems. vol. 20, Curran Associates, Inc., URL https://proceedings.neurips.cc/paper/2007/file/d7322ed717dedf1eb4e6e52a37ea7bcd-Paper.pdf.

Parvaiz, A., Khalid, M.A., Zafar, R., Ameer, H., Ali, M., Fraz, M.M., 2023. Vision transformers in medical computer vision—A contemplative retrospection. Eng. Appl. Artif. Intell. 122, 106126, URL https://www.sciencedirect.com/science/article/pii/S095219762300310X.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 32. Curran Associates, Inc., pp. 8024–8035, URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I., 2021. Learning transferable visual models from natural language supervision. In: Meila, M., Zhang, T. (Eds.), Proceedings of the 38th International Conference on Machine Learning. In: Proceedings of Machine Learning Research, 139, PMLR, pp. 8748–8763, URL https://proceedings.mlr.press/v139/radford21a.html.

Reimers, N., Gurevych, I., 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3982–3992, URL https://aclanthology.org/D19-1410.

Sammut, C., Webb, G.I. (Eds.), 2010. TF–IDF. In: Encyclopedia of Machine Learning. Springer US, Boston, MA, pp. 986–987, URL https://doi.org/10.1007/978-0-387-30164-8_832.

Sharma, R., Gopalani, D., Meena, Y., 2023. An anatomization of research paper recommender system: Overview, approaches and challenges. Eng. Appl. Artif. Intell. 118, 105641, URL https://www.sciencedirect.com/science/article/pii/S0952197622006315.

Shi, W., Wang, H., Lou, X., 2023. Multi-modal graph reasoning for structured video text extraction. Comput. Electr. Eng. 107, 108641, URL https://www.sciencedirect.com/science/article/pii/S0045790623000666.

Shuai, J., Zhang, K., Wu, L., Sun, P., Hong, R., Wang, M., Li, Y., 2022. A review-aware graph contrastive learning framework for recommendation. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '22, Association for Computing Machinery, New York, NY, USA, pp. 1283–1293, URL https://doi.org/10.1145/3477495.3531927.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations. URL https://doi.org/10.48550/arXiv.1409.1556.

Tay, Y., Luu, A.T., Hui, S.C., 2018. Multi-pointer co-attention networks for recommendation. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD '18, Association for Computing Machinery, New York, NY, USA, pp. 2309–2318, URL https://doi.org/10.1145/3219819.3220086.

Valcarce, D., Landin, A., Parapar, J., Barreiro, Á., 2019. Collaborative filtering embeddings for memory-based recommender systems. Eng. Appl. Artif. Intell. 85, 347–356, URL https://www.sciencedirect.com/science/article/pii/S0952197619301599.

Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y., Xiao, T., He, T., Karypis, G., Li, J., Zhang, Z., 2019. Deep graph library: A graph-centric, highly-performant package for graph neural networks. URL https://arxiv.org/abs/1909.01315.

Wang, K., Zhu, Y., Liu, H., Zang, T., Wang, C., 2022. Learning aspect-aware high-order representations from ratings and reviews for recommendation. ACM Trans. Knowl. Discov. Data (2022) Just Accepted, URL https://doi.org/10.1145/3532188.

Wang, K., Zhu, Y., Liu, H., Zang, T., Wang, C., 2023. Learning aspect-aware high-order representations from ratings and reviews for recommendation. ACM Trans. Knowl. Discov. Data 17 (1), URL https://doi.org/10.1145/3532188.

Wu, S., Sun, F., Zhang, W., Xie, X., Cui, B., 2022a. Graph neural networks in recommender systems: A survey. ACM Comput. Surv. (2022) URL https://doi.org/10.1145/3535101.

Wu, S., Sun, F., Zhang, W., Xie, X., Cui, B., 2022b. Graph neural networks in recommender systems: A survey. ACM Comput. Surv. (2023) 55 (5), URL https://doi.org/10.1145/3535101.

Xie, Y., Xu, Z., Zhang, J., Wang, Z., Ji, S., 2023. Self-supervised learning of graph neural networks: A unified review. IEEE Trans. Pattern Anal. Mach. Intell. 45 (2), 2412–2429, URL https://pubmed.ncbi.nlm.nih.gov/35476575/.

Xu, Y., Bin, Y., Wei, J., Yang, Y., Wang, G., Shen, H.T., 2023. Multi-modal transformer with global-local alignment for composed query image retrieval. IEEE Trans. Multimed. 1–13, URL https://ieeexplore.ieee.org/document/10012544.

Xu, K., Hu, W., Leskovec, J., Jegelka, S., 2019. How powerful are graph neural networks? In: International Conference on Learning Representations. URL https://openreview.net/forum?id=ryGs6iA5Km.

Yang, X., Yan, M., Pan, S., Ye, X., Fan, D., 2023. Simple and efficient heterogeneous graph neural network. Proc. AAAI Conf. Artif. Intell. 37 (9), 10816–10824, URL https://ojs.aaai.org/index.php/AAAI/article/view/26283.

Yang, M., Zhou, M., Liu, J., Lian, D., King, I., 2022. HRCF: Enhancing collaborative filtering via hyperbolic geometric regularization. In: Proceedings of the ACM Web Conference 2022. WWW '22, Association for Computing Machinery, New York, NY, USA, pp. 2462–2471, URL https://doi.org/10.1145/3485447.3512118.

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J., 2018. Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD '18, Association for Computing Machinery, New York, NY, USA, pp. 974–983, URL https://doi.org/10.1145/3219819.3219890.

Yu, J., Xia, X., Chen, T., Cui, L., Hung, N.Q.V., Yin, H., 2023. XSimGCL: Towards extremely simple graph contrastive learning for recommendation. IEEE Trans. Knowl. Data Eng. 1–14, URL https://doi.org/10.1109/TKDE.2023.3288135.

Yu, J., Yin, H., Xia, X., Chen, T., Cui, L., Nguyen, Q.V.H., 2022. Are graph augmentations necessary? Simple graph contrastive learning for recommendation. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '22, Association for Computing Machinery, New York, NY, USA, pp. 1294–1303, URL https://doi.org/10.1145/3477495.3531937.

Zhang, M., Chen, Y., 2020. Inductive matrix completion based on graph neural networks. In: International Conference on Learning Representations. URL https://openreview.net/forum?id=ByxxgCEYDS.

Zhang, L., Chen, L., Ou, W., Zhou, C., 2022. Semi-supervised constrained graph convolutional network for cross-modal retrieval. Comput. Electr. Eng. 101, 107994, URL https://www.sciencedirect.com/science/article/pii/S0045790622002622.

Zhang, S., Jiang, Z., Yao, J., Feng, F., Kuang, K., Zhao, Z., Li, S., Yang, H., Chua, T.-s., Wu, F., 2023. Causal distillation for alleviating performance heterogeneity in recommender systems. IEEE Trans. Knowl. Data Eng. 1–16, URL https://ieeexplore.ieee.org/document/10168248.

Zhang, S., Yao, L., Sun, A., Tay, Y., 2019. Deep learning based recommender system: A survey and new perspectives. ACM Comput. Surv. (2020) 52 (1), URL https://doi.org/10.1145/3285029.

Zhao, T., Liu, Y., Neves, L., Woodford, O., Jiang, M., Shah, N., 2021. Data augmentation for graph neural networks. Proc. AAAI Conf. Artif. Intell. 35 (12), 11015–11023, URL https://ojs.aaai.org/index.php/AAAI/article/view/17315.

Zheng, L., Noroozi, V., Yu, P.S., 2017. Joint deep modeling of users and items using reviews for recommendation. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. WSDM '17, Association for Computing Machinery, New York, NY, USA, pp. 425–434, URL https://doi.org/10.1145/3018661.3018665.

Zhu, Q., Yang, C., Xu, Y., Wang, H., Zhang, C., Han, J., 2021. Transfer learning of graph neural networks with ego-graph information maximization. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (Eds.), Advances in Neural Information Processing Systems. vol. 34, Curran Associates, Inc., pp. 1766–1779, URL https://proceedings.neurips.cc/paper/2021/file/0dd6049f5fa537d41753be6d37859430-Paper.pdf.