CAGK: Collaborative Aspect Graph Enhanced Knowledge-based Recommendation

Xiaotong Song[1], Huiping Lin[1] *, Jiatao Zhu[1], Xinyi Gong[1]
[1] Peking University
xt.song@stu.pku.edu.cn, linhp@ss.pku.edu.cn,
jiataozhu@stu.pku.edu.cn, tori.gong@stu.pku.edu.cn

Abstract

Auxiliary information, such as knowledge graph (KG), has become increasingly crucial in recommender systems. However, the current KG-based recommendation still has some limitations: (1) low link rates between items and KG entities, (2) redundant knowledge in KG. In this paper, we introduce the aspect, which refers to keywords describing item attributes in reviews, to KG-based recommendation, and propose a new model, ***C**ollaborative **A**spect **G**raph Enhanced **K**nowledge-based Network* (CAGK). Firstly, CAGK builds a Collaborative Aspect Graph (CAG) with user-item interactions, aspects and KG, where aspects can fill most of the sparsity. Secondly, we leverage interactive information and aspect features to generate aspect-aware guidance signals to customize knowledge extraction and eliminate redundant knowledge. Lastly, we utilize low ratings and negative aspect sentiment to capture features that users dislike to prevent repetitive recommendations of disliked items. Experimental results on two widely used benchmark datasets, Amazon-book and Yelp2018, confirm the superiority of CAGK.

Keywords: Aspect, Knowledge Graph, Recommender Systems

1. Introduction

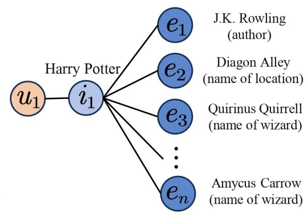
A recommender system (RecSys) serves as a powerful tool to provide personalized recommendation solutions for a range of domains, such as E-commerce platforms, search engines, and social networks. Its main purpose is to address the issue of information overload(Sarwar et al., 2001). One widely successful approach is Collaborative Filtering (CF). CF recommends relevant items based on the assumption that users with similar behaviors share similar item preferences (Koren et al., 2009). However, CF-based methods often encounter challenges such as data sparsity and cold-start (Cheng et al., 2018; McAuley and Leskovec, 2013; Wang et al., 2018c). To overcome these limitations, some studies propose integrating auxiliary information such as social networks, user or item attributes, images, contextual data, etc.

Among various auxiliary information, the Knowledge Graph (KG) contains rich entities and relations, showing great potential in improving recommendation accuracy and interoperability, such as Freebase(Bollacker et al., 2008) and YAGO (Suchanek et al., 2007). With the help of KG, the problem of data sparsity as well as cold start can be alleviated. However, simply integrating KG into the recommendation model does not necessarily improve performance (Chen et al., 2022), and there are mainly the following problems:

(1) **Link sparsity.** Most KG-based methods link items to KG entities through item titles. Figure 1 (a) shows the linkage ratio of the widely used link method KB4Rec (Zhao et al., 2019), which refers to the ratio of items linked to the side information (here refers to the KG entity) among all items. We can see that the linkage ratio between the recommendation dataset Amazon-book and the knowledge graph Freebase is only 4.7%, and other link linkage ratios are also very low, which shows that the link between items and KG is very sparse. Therefore, the help of the knowledge graph is limited.

Linkage ratio	Amazon-book	LFM-1b
Freebase	4.7%	19.4%
YAGO	0.8%	0.8%

(a) Link sparsity.



(b) Knowledge redundancy.

Figure 1: Illustration of link sparsity and knowledge redundancy.

(2) **Knowledge redundancy.** The information in KG may not all be helpful for recommendation. As shown in Figure 1(b), item i_1, named *Harry Potter*, is linked to multiple entities from the knowledge graph, where the entity e_1, named *J.K. Rowling*, is a famous author. Therefore, the book *Harry Potter* can be linked to other books written by *J.K. Rowling*, thus improving the recommendation accuracy and interpretability. However, entity e_2, e_3 and e_n are names of locations and wizards in *Harry Potter*. The other entities that e_2, e_3 and e_n can link to are basically limited to *Harry Potter*, so there is no

*Corresponding author

LREC-COLING 2024, pages 2612–2621
20-25 May, 2024.

obvious effect on recommendation. In addition, in the process of information propagation, too many entities linked to i_1 will weaken the interaction information. Therefore, it's necessary to effectively utilize the knowledge from KG.

Reviews contain rich semantic information and are very valuable auxiliary information. Extracting aspects from reviews using natural language processing (NLP) tools has become a standard approach to enhance RecSys. ANR (Chin et al., 2018) designed a co-attention mechanism to evaluate the importance of different aspects jointly. AARM (Guan et al., 2019) captured the importance of item-specific aspects by the neural attention network. Although these methods achieve progressive results, they still have many limitations, 1) using topic models to extract aspects, which are coarse-grained and of low quality, and 2) failing to discover the advantages of aspects for the KG-based recommendation. Before further analysis, let's establish a clear distinction between the aspect and entity. The definitions of the two are as follows:

- **Aspect**: Contains high-level semantics, representing keywords about item attributes mentioned by users in reviews, such as "*narrative style*" and "*subject matter*" from book reviews.

- **Entity**: Refers to the representation of various things that exist in the real world, such as "*Harry Potter*", "*KFC*", etc.

It can be found that they have the following differences: (1) The information expressed by the aspect is more personalized and can more accurately reflect the user's personalized preferences. (2) The entity in KG represents authoritative and extensive information and establishes high-level connections between items. It can be observed that the information contained in aspects and entities can be complementary, so it is meaningful to enhance KG-based recommendation with aspects.

Aspects have many advantages for KG-based recommendation. Firstly, the problem of link sparsity can effectively be alleviated by the fact that most items have reviews that correspond to a set of aspects. Secondly, aspects contain personalized user preferences. Signals generated based on aspects can guide the model to extract useful information from KG and filter redundancy. Finally, we extract negative sentiment information, as a source of evidence for not recommending.

In this paper, we focus on exploring the potential of aspects in KG-based recommendation. We first extract aspects using Pre-trained Language Models (PLMs). Therefore, the extracted aspects are of higher quality. In addition, we also use PLMs to analyze the sentiment (positive or negative) of the aspect. Then we propose a novel model named *Collaborative Aspect Graph enhanced Knowledge-based Network* (CAGK), which has three main modules. 1) Aspect-aware graph aggregation, which uses aspects to model user preferences and item characteristics. 2) Knowledge extraction with aspect-aware collaborative guidance, which employs an aspect-aware attention mechanism to learn the weight of each neighbor of the item in KG and obtain the embedding of the item on the KG side through attentive propagation. 3) Negative sentiment modeling, which inputs the negative graph into the MLP network to obtain the negative representations of users and items. Our contributions are summarized as follows:

- As far as we know, we are the first to exploit the potential of aspects in KG-based recommendation.

- We propose a novel framework named Collaborative Aspect Graph enhanced Knowledge-based Network (CAGK), which alleviates the challenges posed by sparse links and redundant knowledge and makes full use of the negative sentiment information.

- Extensive experiments on two widely used benchmark datasets demonstrate the superiority of our method.

2. Related work

In this section, we introduce the related work of CAGK, which includes KG-based methods and aspect-based methods.

2.1. KG-based methods

The KG-based recommendation can be roughly divided into three categories: embedding-based, path-based, and graph neural network (GNN)-based methods. Embedding-based methods (Ai et al., 2018; Cao et al., 2019; Huang et al., 2018) mainly focus on first-order connectivity (i.e., user-item pairs of interaction data and triples in KG), using KG embedding techniques (such as TransE (Bordes et al., 2013)) to learn entity embeddings. Path-based methods (Catherine and Cohen, 2016; Hu et al., 2018; Ma et al., 2019) demonstrate long-range connectivity by extracting paths through KG entity connections between the target user and item nodes. Then, these paths are used to predict user preferences. GNN-based methods (Jin et al., 2020; Wang et al., 2019a,a) are built on the information aggregation mechanism of graph neural networks (He et al., 2020). Typically, it combines information from one-hop nodes to update the representation of the center node. After recursively executing such propagation, information from multi-hop nodes can be encoded in the representation. Therefore, this method can model distant connections. KGAT (Wang et al., 2019b) combines user-item interactions and KG into a heterogeneous graph and applies aggregation mechanisms to it.

2.2. Aspect-based method

In recent years, extensive research has been conducted on utilizing aspects to improve the performance and interpretability of recommender systems (Chin et al., 2018). These methods can be classified into the following two groups. The first group attempts to extract aspects based on existing sentiment analysis NLP tools (Otter et al., 2020). Then, the obtained aspect representations are merged into a matrix factorization framework to obtain more accurate recommendations. For example, EFM (Zhang et al., 2014) and MTER (Wang et al., 2018b) adopt phrase-level NLP tools for aspect-level sentiment extraction. The second group designs specific internal components such as topic modeling to automatically learn interpretable aspect representations for users and items from reviews (Cheng et al., 2018). In particular, aspect-aware extraction of semantic information from reviews is achieved using these internal components. In summary, compared with the first category, aspect-aware methods can extract high-level semantic features from reviews, and provide improved results and interpretability.

3. Task Formulation

This section mainly explains the construction process of the collaborative aspect graph (CAG), which is the main input of the model.

Collaborative Aspect Graph (CAG). First, the user-item interaction graph is defined as $G_{UI} = \{(u, y_{ui}, i)|u \in \mathcal{U}, i \in \mathcal{I}\}$, where $\mathcal{U}$ and $\mathcal{I}$ represent the user set and item set respectively, $y_{ui} \geq 0$ means that there is an observed interaction between user u and item i, and the score is y_{ui}; otherwise $y_{ui} = -1$. Considering the relationship between users, items and aspects, we construct $G_{AG} = \{(u, p, a), (i, p, a)|u \in \mathcal{U}, i \in \mathcal{I}, a \in \mathcal{A}\}$, where $\mathcal{A}$ indicates the set of aspects, and $p \in \{1, -1\}$ indicates whether the aspect is positive (1) or negative (-1) for a certain user/item. As for the KG, we construct $G_{KG} = \{(h, r, t)|h, t \in \mathcal{E}, r \in \mathcal{R}\}$, where each triple describes a relationship r between the head entity h and the tail entity t. Finally, the unified graph $G = \{(w, r, v)|w, v \in \mathcal{E}', r = \mathcal{R}'\}$, where $\mathcal{E}' = \mathcal{E} \cup \mathcal{U} \cup \mathcal{I} \cup \mathcal{A}, \mathcal{R}' = \mathcal{R} \cup \{y\} \cup \{p\}$. We divide the CAG into positive (CAG-P) and negative (CAG-N) graphs according to the value of the relations. Specifically, the edge with $p = 1$ in G_{AG}, the edge with $y_{ui} >= 3$ in G_{UI} and its associated nodes form the CAG-P; the edge with $p = -1$ in G_{AG}, the edge with $y_{ui} < 3$ in G_{UI} and its associated nodes belong to the CAG-N; the edges and nodes in the G_{KG} have no emotional tendency, so they will appear in both the CAG-P and CAG-N.

4. Method

We now present the proposed Collaborative Aspect Graph enhanced Knowledge-based Network (CAGK). The calculation process of CAG-P is illustrated in Figure 2.

4.1. Aspect-aware Graph Aggregation

Based on the CAG-P, we start computing the representation of users and items under the Graph neural Network (GNN) paradigm. With the aspect involved, users with similar preferences will not only be connected through the items they have both interacted with but also through the aspects they have reviewed together, the same is true for items. Taking restaurant recommendation as an example, the aspects mentioned by the user reflect what the user values, such as *environment*, *dishes*, etc. In addition, users with strict requirements for the environment will generate high-order connections with places with good environments, as they are connected to the same aspect. Therefore, in this module, the user and item representation are enhanced by aspects.

4.1.1. Dual-grained user preferences learning:

We first extract coarse-grained collaboration information from G_{UI}. Considering a user u, we use $N_u^i = \{i|(u, y_{ui}, i) \in G_{UI}\}$ to denote the neighbors of u on G_{UI}. Technically, the user u's item side representation in the first layer can be calculated as:

$$e_{ui}^{(1)} = \frac{1}{|N_u^i|} \sum_{i \in N_u^i} e_r \odot e_i^{(0)}, \tag{1}$$

where $e_{ui}^{(1)} \in \mathbb{R}^d$ is the representation of u in layer 1; $e_i^{(0)} \in \mathbb{R}^d$ and $e_r \in \mathbb{R}^d$ is the ID embedding of item i and relation r.

We then aggregation user u's neighbors on G_{AG} to create the fine-grained representation of user u as:

$$e_{ua}^{(1)} = \sum_{a \in N_u^a} \alpha(u, a) \mathrm{e}_a, \tag{2}$$

where $e_{ua}^{(1)} \in \mathbb{R}^d$ is the representation of u from aspects in layer 1. $N_u^a = \{a|(u, p, a) \in G_{AG}\}$ is the set of aspects that are adjacent to the user u. Considering users have different degrees of preferences for different aspects, so we introduce an attention

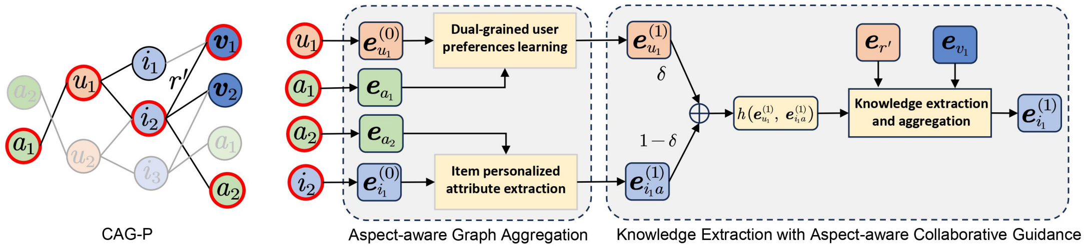


Figure 2: Framework diagram of the processing flow of CAG-P in layer $l = 1$ of CAGK. We illustrate with u_1, i_2, a_1, a_2 and v_1 as examples. The figure highlights the proposed collaborative aspect graph, the aspect-aware graph aggregation module and the knowledge extraction with aspect-aware collaborative guidance module.

score $\alpha(u, a)$ to differentiate the importance of aspects as:

$$\alpha(u, a) = \frac{\exp(e_a^T e_u^{(0)})}{\sum_{a' \in N_u^a} \exp(e_{a'}^T e_u^{(0)})}. \tag{3}$$

Through the information propagation at the above coarse-grained and fine-grained levels, we can obtain the user's dual-grained aggregation representation:

$$e_u^{(1)} = e_{ui}^{(1)} + e_{ua}^{(1)}, \tag{4}$$

where $e_u^{(1)}$ denotes the user u's first-order neighbor aggregation.

4.1.2. Item personalized attribute extraction

To address the problem of insufficient representation of items caused by the sparsity of KG-based recommendations, we obtain the representation of items by aggregating aspects on G_{AG}:

$$e_{ia}^{(1)} = \sum_{a \in N_i^a} \alpha(i, a) \boldsymbol{e}_a, \tag{5}$$

where $e_{ia}^{(1)}$ is the representation of item i from aspects, and $\alpha(i, a)$ is the attention score to distinguish the importance of different aspects on reflecting item characteristics and the calculation process is shown in eq. 4.

4.2. Knowledge Extraction with Aspect-aware Collaborative Guidance

For a target item i, CAGK needs to further extract knowledge from the KG. In order to filter noise in the KG, we first need to generate a guidance signal.

4.2.1. Aspect-aware Guidance Encoding

Based on the embeddings of target user u ($e_u^{(1)}$) and target item i ($e_{ia}^{(1)}$) from eq. 4 and 5, we can encode them to a guidance signal $h(\boldsymbol{e}_u, \boldsymbol{e}_{ia}) : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ as:

$$h(\boldsymbol{e}_u^{(1)}, \boldsymbol{e}_{ia}^{(1)}) = \delta \boldsymbol{e}_u^{(1)} + (1 - \delta) \boldsymbol{e}_{ia}^{(1)}, \tag{6}$$

where $\delta \in (0, 1)$ is a trainable weight.

4.2.2. Knowledge extraction and aggregation

We use aspect-aware guidance to filter the noise on G_{KG}. According to the CAG-P in Figure 2, given i's first-order connectivity in G_{KG} $N_i^v = \{v | (i, r', v) \in G_{KG}\}$, we define the notation $\{\langle u, i, a \rangle, i, r', v\}$ to represents that (i, r', v) is guided by the target triplet $\langle u, i, a \rangle$. We first get the general relation embedding, and then compute the customized embedding with the guidance as follow:

$$\boldsymbol{e}_{r'}^{\langle u,i,a \rangle} = h(\boldsymbol{e}_u^{(1)}, \boldsymbol{e}_{ia}^{(1)}) \odot \boldsymbol{e}_{r'}, \tag{7}$$

where $\odot$ represents the element-wise product. By fusing the aspect-aware guidance, $\langle u, i, a \rangle$ can simultaneously capture the relational representation of relation r', the aspect as well as the collaborative information of u and i. Based on this, we calculate the attention weight of item i to entity v as:

$$\begin{aligned} &\beta(\langle u, i, a \rangle, i, r', v) \\ &= (\mathbf{W} \boldsymbol{e}_v^{(0)})^T \tanh(\mathbf{W} \boldsymbol{e}_i^{(0)} + \boldsymbol{e}_{r'}^{<u,i,a>}), \end{aligned} \tag{8}$$

where $\mathbf{W} \in \mathbb{R}^{d \times d}$ is the trainable transformation matrix that casts $\boldsymbol{e}_i^{(0)}$ and $\boldsymbol{e}_v^{(0)}$ into relation r''s latent space; and $\tanh$ is the nonlinear activation function. Hereafter, we normalize the attention score by adopting the softmax function as:

$$\begin{aligned} &\beta'(\langle u, i, a \rangle, i, r', v) \\ &= \frac{\exp(\beta(\langle u, i, a \rangle, i, r', v))}{\sum_{(i, r^*, v^*) \in N_i^v} \exp(\beta(\langle u, i, a \rangle, i, r^*, v^*))}. \end{aligned} \tag{9}$$

Then we move on to the computation of the latent representation of i's neighboring entities as follows:

$$\boldsymbol{e}_{iv}^{(1)} = \sum_{(i, r', v) \in N_i^v} \beta'(\langle u, i, a \rangle, i, r', v) \boldsymbol{e}_v^{(0)}. \tag{10}$$

Combining the item representation obtained in eq.5, we can obtain the information aggregation of the item's first-order neighbors:

$$\boldsymbol{e}_i^{(1)} = \boldsymbol{e}_{ia}^{(1)} + \boldsymbol{e}_{iv}^{(1)}. \tag{11}$$

4.3. Negative Sentiment Modeling

In this section, we extract user's negative sentiment information on items and aspects in CAG-N. Since the information propagation mechanism in GNNs is not suitable for negative graphs (Seo et al., 2022), we input the initialized user and item representations into multi-layer perceptron (MLP) as follows:

$$\boldsymbol{e}_u^N, \boldsymbol{e}_i^N = MLP(G_{CAG-N}), \tag{12}$$

where $\boldsymbol{e}_u^N, \boldsymbol{e}_i^N \in \mathbb{R}^d$ separately denote the negative representations of user u and item i. We apply the negative representation to the model prediction module, which will be explained in section 4.4.

4.4. Model Prediction

After L layers on the positive graph CAG-P, we get the representations of user u and item i at each layer, and then add them together as the final representations:

$$\boldsymbol{e}_u^P = \boldsymbol{e}_u^{(0)} + \cdots + \boldsymbol{e}_u^{(L)}, \boldsymbol{e}_i^P = \boldsymbol{e}_i^{(0)} + \cdots + \boldsymbol{e}_i^{(L)}. \tag{13}$$

Thereafter, we use the inner product of user u and the item i's positive representations to reflect the similarity between user preferences and item characteristics, and the negative representations to reflect the similarity between user dislike and item defects. Finally, we calculate the score as follows:

$$\hat{y}(u,i) = \boldsymbol{e}_u^P \boldsymbol{e}_i^P - \boldsymbol{e}_u^N \boldsymbol{e}_i^N + 1. \tag{14}$$

Add 1 because of the possibility of negative values.

4.5. Model Optimization

$$L_{CAGK} = \sum_{(u,i,j) \in T} -ln\sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda \left\|\Theta\right\|_2^2 \tag{15}$$

where Θ is the set of model parameters; and $\|\Theta\|_2^2$ is the L2-regularization parameterized controlled by the hyperparameters λ to avoid over-fitting.

5. Experiments

5.1. Dataset

In order to verify the effect of our proposed method on different datasets, two types of real-world review datasets are used in the experiment: Amazon-book and Yelp2018. The statistics of the two datasets are shown in Table 1.

		Amazon-book	**Yelp2018**
Interaion	# users	70,679	45,919
	# items	24,915	45,538
	# interactions	847,733	1,185,068
KG	# entities	88,572	90,961
	# relations	39	42
	# triples	2,557,746	1,853,704
Aspect	# aspects	207734	190521
	# linkage ratio	96.3%	98.4%

Table 1: Statistics of the Datasets

Amazon book: Amazon-review is a widely used recommendation dataset (He and McAuley, 2016). We chose Amazon-book from it. To ensure the quality of the dataset, we keep users and items with at least 10 interactions.

Yelp2018: This dataset is from the 2018 edition of the Yelp Challenge. Here, we consider local businesses like restaurants, bars, etc. as items. Similarly, we use a 10-core setting to ensure quality.

In addition to user-item interactions, we need to build a knowledge graph for each dataset. For amazon-book, we adopted the method in KB4Rec (Zhao et al., 2019), where items are mapped to Freebase by title matching. For Yelp2018, we follow the approach in KGAT (Wang et al., 2019b). To ensure knowledge graph quality, we then filter out uncommon entities (i.e., less than 10 in both datasets) and retain relations that appear in at least 50 triples.

We use the PLM InstructABSA (Scaria et al., 2023) to extract aspects and perform sentiment classification on them. In order to ensure the quality, we set the maximum number of user or item connection aspects to 20, and filter out the aspects with occurrences less than 5 aspects. Then we organize the results into user aspect sets and item aspect sets according to the interaction relationship. After preprocessing the interaction, knowledge graph and aspects of the two datasets, the statistical results are summarized in Table 1. It shows that the linkage ratio of aspects (side information) with items are above 95% in both datasets, which is much higher than the link ratio of items to KG entities, indicating that aspect can alleviate the sparsity problem.

5.2. Baselines

This section briefly introduces the main comparison models and methods in the following experiments: including the CF-based method NFM, the embedding-based method CKE, the graph neural network-based methods RippleNet, KGAT, aspect-based methods ANR, CARP, knowledge-optimized recommendation methods CG-KGR and SiReN that distinguishes between positive and negative

Datasets	Model	recall@10	recall@20	ndcg@10	ndcg@20
Amazon-book	NFM	0.1266	0.1366	0.0794	0.0913
	CKE	0.1264	0.1342	0.0522	0.0698
	RippleNet	0.1256	0.1336	0.0796	0.091
	KGAT	0.1325	0.1489	0.0833	0.1006
	KGIN	<u>0.1441</u>	<u>0.1687</u>	<u>0.0875</u>	<u>0.1095</u>
	ANR	0.124	0.132	0.0727	0.0801
	CARP	0.1292	0.1441	0.0801	0.0923
	CG-KGR	0.1299	0.145	0.0859	0.1079
	SiReN	0.1175	0.1302	0.0823	0.0986
	CAGK	**0.1501**	**0.1788**	**0.092**	**0.1204**
Yelp2018	NFM	0.048	0.066	0.0591	0.081
	CKE	0.0479	0.0657	0.058	0.0805
	RippleNet	0.0484	0.0664	0.057	0.0858
	KGAT	0.0543	0.0712	0.0602	0.0867
	KGIN	<u>0.0588</u>	<u>0.0754</u>	<u>0.0668</u>	<u>0.088</u>
	ANR	0.0497	0.0672	0.0524	0.0771
	CARP	0.0521	0.0703	0.0598	0.0813
	CG-KGR	0.0513	0.0698	0.0633	0.0847
	SiReN	0.0574	0.074	0.0559	0.0792
	CAGK	**0.0622**	**0.0814**	**0.0717**	**0.0943**

Table 2: Performance comparison between the baselines and our model CAGK. The best score is bold and the second-best score is underlined in each row.

sentiment.

- **NFM** (He and Chua, 2017): A state-of-the-art factorization model that incorporates frequency modulation networks into neural networks.
- **CKE** (Zhang et al., 2016): A typical regularization-based approach that leverages TransR-derived semantic embeddings to enhance matrix factorization.
- **RippleNet** (Wang et al., 2018a): A model combines regularization-baesd and path-based approaches. User representations are enriched by adding items to each user's root path.
- **KGAT** (Wang et al., 2019b): A state-of-the-art GNN-based recommender system. It applies an attention-based neighborhood aggregation mechanism on the overall graph to combine KG with the user-item graph to generate user and item representations. User-item relations and KG relations are used as attention weights in the adjacency matrix.
- **KGIN** (Wang et al., 2021): Building on KGAT, identifying user-item relationships at the fine-grained level of intent, exploiting auxiliary item knowledge to explore the intent behind user-item interactions, and exploiting relational dependencies to preserve the semantics of long-range connections.
- **ANR** (Chin et al., 2018): An aspect-based recommendation model for estimating the rank of latent aspects and the importance of latent aspects. The latent aspect score is derived by the weighted sum of all words embedded in the review. The latent aspect importance is inferred by using the shared similarity between each pair of latent aspects of user-item. Finally, an overall rating is inferred for any user-item pair by combining their associated values.
- **CARP** (Li et al., 2019): A capsule network-based user review rating prediction model is proposed, which uses three points of user opinion, item, and emotion to reason, and constructs positive and negative double capsules, which can judge the degree of emotion by the length of the capsule, which can be more good understanding of user ratings.
- **CG-KGR** (Chen et al., 2022): Propose a knowledge-aware recommendation model that enables rich and coherent learning of KGs and user-item interactions through a collaborative guidance mechanism.
- **SiReN** (Seo et al., 2022): Model negative sentiment interaction information in a graph neural network-based method, analyze the difficulties faced by the graph neural network in processing negative sentiment information, and finally

use the linear layer to extract the negative representation of users and items for recommendation.

5.3. Experiment Setting

In terms of experimental datasets division, the training set, verification set, and test set are randomly divided according to 8:1:1. The commonly used metrics including Recall and NDCG are leveraged for comparing different models. The random seed remains fixed to ensure consistency when compared with different models. In terms of experimental execution, CAGK is based on the framework PyTorch and the graph deep learning framework DGL, repeats each model 5 times on each dataset and takes the average as the final experimental result. For each user and item, the maximum number of connected aspects K is selected in [5,10,15,20] for debugging. The range of the number of message propagation layers L is set to [1,2,3,4]. The batch size is tuned amongst [256,512,1024,2048]. For the neighbor sampling mechanism in the model, the sampling number is selected in [5, 10, 15, 20]. In the experiment, the above important hyperparameters were debugged within the specified range. Finally, we use the Adam gradient descent optimization algorithm (Kingma and Ba, 2014) for computing the loss function that minimizes the objective.

5.4. Experiment Results

This section compares the performance of our proposed CAGK with the baselines on two benchmark datasets. The experimental comparison results are shown in Table 2. By analyzing the above experiments, the following conclusions can be drawn:

- CAGK consistently achieves the best performance on all datasets and metrics. In particular, in the Amazon-book and Yelp2018 datasets, CAGK improves recall@20 by 5.99% and 7.96% compared to the state-of-the-art KGIN. By combining aspects to build a collaborative aspect graph, fine-grained user preferences and item features can be captured, thereby effectively supplementing recommendation information, which verifies the effectiveness of aspect information and models.
- SL method NFM has achieved better performance than CKE and RippleNet, indicating that some KG-based methods have achieved poor results because the knowledge is not fully utilized and the noise is not filtered. In particular, CKE uses a regularization-based model, and RippleNet uses a path-based model. It can be seen that the key to poor performance is not the model, but the sparseness and noise of the knowledge graph.

method	Amazon-book		Yelp2018	
	recall@20	ndcg@20	recall@20	ndcg@20
w/o AGA	0.1712	0.1107	0.0736	0.0891
w/o KEACG	0.1754	0.1148	0.0786	0.0882
w/o NSM	0.1769	0.1155	0.0793	0.0933
w/o PLMs	0.1673	0.1050	0.0722	0.0871
CAGK	**0.1788**	**0.1204**	**0.0814**	**0.0943**

Table 3: Ablation Study

- CAGK performs better than KGAT and KGIN, This confirms the effectiveness of the aspect. Specifically, it proves that the aspect-aware guidance signal can filter out redundant knowledge. In addition, compared with KGIN, CAGK replaces the implicit intent in KGIN with explicit aspects, which are more personalized.
- Compared with the aspect-based methods ANR and CARP, the advantages of CAGK are reflected in three angles. First, it uses a more advanced PLM for aspect extraction, which is more accurate; second, it incorporates knowledge, so that more extensive information can be integrated; third, the aspects are divided into positive sentiment and negative sentiment, which are modeled separately.
- CG-KGR optimizes the recommendation model based on knowledge graphs. Compared to CG-KGR, CAGK uses not only interactive information but also aspect information to obtain the guidance signal. Therefore, the effect of CAGK is significantly better than CG-KGR.
- SiReN distinguishes positive sentiment and negative sentiment of interaction according to the level of scores. CAGK distinguishes positive and negative sentiment at the aspect level, which is an extension. In addition, CAGK also combines knowledge and aspect information, so higher performance can be obtained.

5.5. Ablation Study

In order to fully analyze the performance of CAGK, this section explores the influence of the key modules Aspect-aware Graph Aggregation (AGA), Knowledge Extraction with Aspect-aware Collaborative Guidance (KEACG) and Negative Sentiment Modeling (NSM).

- From table 3, it can be seen that after removing the AGA module, there is a clear decrease in the recall and ndcg, which verifies the effectiveness of the way aspect features are extracted in AGA. The results of removing aspects are still better than KGAT and KGIN because CAGK generates a guidance module

	Amazon-book		Yelp2018	
	recall@20	ndcg@20	recall@20	ndcg@20
CAGK-1	0.1613	0.1103	0.0771	0.0911
CAGK-2	0.1705	0.1127	0.0806	0.0927
CAGK-3	0.1779	0.1132	0.0804	0.0925
CAGK-4	**0.1788**	**0.1204**	**0.0814**	**0.0943**

Table 4: Effect of embedding propagation layer number (L).

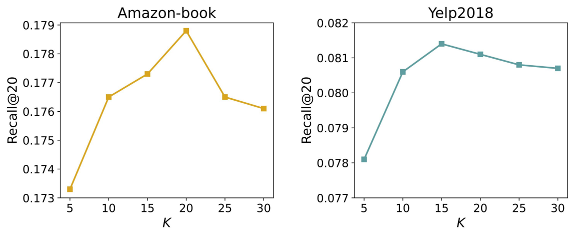


Figure 3: Effect of the number of aspects sampled (K).

for knowledge extraction, which can filter the noise in knowledge.

- After deleting the knowledge extraction module KEACG, both metrics drop, indicating that the module's filtering of knowledge is meaningful. KEACG can effectively filter out customized knowledge and eliminate noise. In addition, compared with CG-KGR, the model that removes the aspect-guided knowledge extraction module still has advantages, mainly because we introduce aspect information.

- Comparing the w/o NSM model with the CAGK, it can be found that distinguishing between positive and negative sentiment and using negative sentiment information can improve the recommendation performance. Extracting The characteristics of negative sentiment information can grasp what items users don't like, or what aspects of items they value more, so as to make more accurate recommendations.

- Comparing CAGK with w/o PLMs method, it can be seen that the performance of using the topic model to extract aspects is obviously inferior to the method using PLMs (CAKG), thus indicating that the aspect quality obtained from PLM exceeds the aspect quality of the topic model.

The above ablation experiments further verified the effectiveness of the three innovative modules proposed in this paper and confirmed the advantages of CAGK.

5.6. Hyperparameter Analysis

In this section, we will conduct experiments on the important hyper-parameters in the model: the number of model layers L and the number of aspect samples K.

- Effect of the number of model layers L: We investigate the efficiency of using multiple embedding propagation layers by varying L in the range [1,2,3,4]. We use CAGK-1 for models with one layer and similar notation for models with other layers. We summarize the results in Table 4, It can be observed that increasing the depth of CAGK can greatly improve performance. Clearly, CAGK-2, CAGK-3, and CAGK-4 achieve consistent improvements over CAGK-1 in all aspects. We attribute this improvement to effectively modeling high-order relationships among users, items, aspects and entities.

- Effect of aspect sampling number K: Figure 3 shows the results under different K value settings. It can be seen that when the number of sampled aspects is too small, the aspect information that can be transmitted to the model is limited, resulting in the model being unable to fully learn fine-grained user preferences and item attributes, thereby affecting the performance. When the number of sampled aspect K is too large, there are too many aspect nodes linked by users or items, and these aspects are not necessarily useful for recommendation, so a large number of repeated nodes and edges will appear, which will bring noise to the model. For the datasets Amazon-book and Yelp2018, the best results can be obtained by taking 20 and 15 aspect sampling numbers K respectively.

6. CONCLUSION

In this paper, we use aspects extracted by PLMs to solve two existing problems in the KG-based RecSys and design a novel framework CAGK with three important modules: Aspect-aware Graph Aggregation, Knowledge Extraction with Aspect-aware Collaborative Guidance and Negative Sentiment Modeling. Experimental results on two widely used benchmark datasets, Amazon-book and Yelp2018, show the superiority of CAGK. This paper opens the door to incorporating aspect information extracted by advanced pre-trained large models into RecSys. There is still much room left for the following work. For example, introducing aspects to improve the diversity of RecSys.

7. Bibliographical References

Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. 2018. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms*, 11(9):137.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. 2019. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *The world wide web conference*, pages 151–161.

Rose Catherine and William Cohen. 2016. Personalized recommendations using knowledge graphs: A probabilistic logic programming approach. In *Proceedings of the 10th ACM conference on recommender systems*, pages 325–332.

Yankai Chen, Yaming Yang, Yujing Wang, Jing Bai, Xiangchen Song, and Irwin King. 2022. Attentive knowledge-aware graph convolutional networks with collaborative guidance for personalized recommendation. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 299–311. IEEE.

Zhiyong Cheng, Ying Ding, Lei Zhu, and Mohan Kankanhalli. 2018. Aspect-aware latent factor model: Rating prediction with ratings and reviews. In *Proceedings of the 2018 world wide web conference*, pages 639–648.

Jin Yao Chin, Kaiqi Zhao, Shafiq Joty, and Gao Cong. 2018. Anr: Aspect-based neural recommender. In *Proceedings of the 27th ACM International conference on information and knowledge management*, pages 147–156.

Xinyu Guan, Zhiyong Cheng, Xiangnan He, Yongfeng Zhang, Zhibo Zhu, Qinke Peng, and Tat-Seng Chua. 2019. Attentive aspect modeling for review-aware recommendation. *ACM Transactions on Information Systems (TOIS)*, 37(3):1–27.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517.

Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 355–364.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 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*, pages 639–648.

Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S Yu. 2018. Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1531–1540.

Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y Chang. 2018. Improving sequential recommendation with knowledge-enhanced memory networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 505–514.

Jiarui Jin, Jiarui Qin, Yuchen Fang, Kounianhua Du, Weinan Zhang, Yong Yu, Zheng Zhang, and Alexander J Smola. 2020. An efficient neighborhood-based interaction model for recommendation on heterogeneous graph. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 75–84.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Chenliang Li, Cong Quan, Li Peng, Yunwei Qi, Yuming Deng, and Libing Wu. 2019. A capsule network for recommendation and explaining what you like and dislike. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 275–284.

Weizhi Ma, Min Zhang, Yue Cao, Woojeong Jin, Chenyang Wang, Yiqun Liu, Shaoping Ma, and Xiang Ren. 2019. Jointly learning explainable rules for recommendation with knowledge graph. In *The world wide web conference*, pages 1210–1221.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172.

Daniel W Otter, Julian R Medina, and Jugal K Kalita. 2020. A survey of the usages of deep learning for natural language processing. *IEEE transactions on neural networks and learning systems*, 32(2):604–624.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295.

Kevin Scaria, Himanshu Gupta, Saurabh Arjun Sawant, Swaroop Mishra, and Chitta Baral. 2023. Instructabsa: Instruction learning for aspect based sentiment analysis. *arXiv preprint arXiv:2302.08624*.

Changwon Seo, Kyeong-Joong Jeong, Sungsu Lim, and Won-Yong Shin. 2022. Siren: Sign-aware recommendation using graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706.

Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018a. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 417–426.

Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019a. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 968–977.

Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. 2018b. Explainable recommendation via multi-task learning in opinionated text data. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 165–174.

Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019b. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 950–958.

Xiang Wang, Xiangnan He, Fuli Feng, Liqiang Nie, and Tat-Seng Chua. 2018c. Tem: Tree-enhanced embedding model for explainable recommendation. In *Proceedings of the 2018 world wide web conference*, pages 1543–1552.

Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, and Tat-Seng Chua. 2021. Learning intents behind interactions with knowledge graph for recommendation. In *Proceedings of the web conference 2021*, pages 878–887.

Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 353–362.

Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 83–92.

Wayne Xin Zhao, Gaole He, Kunlin Yang, Hongjian Dou, Jin Huang, Siqi Ouyang, and Ji-Rong Wen. 2019. Kb4rec: A data set for linking knowledge bases with recommender systems. *Data Intelligence*, 1(2):121–136.