




## Article

# ASKAT: Aspect Sentiment Knowledge Graph Attention Network for Recommendation

Yachao Cui <sup>1,2,3</sup> , Peng Zhou <sup>1</sup> , Hongli Yu <sup>1</sup>, Pengfei Sun <sup>1</sup>, Han Cao <sup>1,\*</sup> and Pei Yang <sup>2,3,\*</sup> 

<sup>1</sup> School of Computer Science, Shaanxi Normal University, Xi'an 710119, China; cuiyachao@snnu.edu.cn (Y.C.); \_zhoupeng@snnu.edu.cn (P.Z.); yuhongli@snnu.edu.cn (H.Y.); sunpf@snnu.edu.cn (P.S.)

<sup>2</sup> Department of Computer Technology and Applications, Qinghai University, Xining 810016, China

<sup>3</sup> Qinghai Provincial Key Laboratory of Media Integration Technology and Communication, Xining 810099, China

\* Correspondence: caohan@snnu.edu.cn (H.C.); yangpeinmgdx@sina.com (P.Y.)

**Abstract:** In modern online life, recommender systems can help us filter unimportant information. Researchers of recommendation algorithms usually utilize historical interaction data to mine potential user preferences. However, most existing methods use rating data to mine user interest preferences, ignoring rich textual information such as reviews. Although some researchers have attempted to combine ratings and reviews for recommendation, we believe the following shortcomings still exist. First, existing methods are overly dependent on the accuracy of external sentiment analysis tools. Second, existing methods do not fully utilize the features extracted from reviews. Further, existing methods focus only on the aspects that users like, while ignoring the aspects that users dislike, and they cannot completely model users' true preferences. To address the above issues, in this paper, we propose a recommendation model based on the aspect of the sentiment knowledge graph attention network (ASKAT). We first use the improved aspect-based sentiment analysis algorithm to extract aspectual sentiment features from reviews. Then, to overcome the difficulty in underutilizing the information extracted from the comments, we build aspects of sentiment-enhanced collaborative knowledge mapping. After that, we propose a new graph attention network that uses sentiment-aware attention mechanisms to aggregate neighbour information. Finally, our experimental results on three datasets, Movie, Amazon book, and Yelp, show that our model consistently outperforms the baseline model in two recommendation scenarios, click-through-rate prediction and Top-k recommendation. Compared with other models, the method shows significant improvement in both recommendation accuracy and personalised recommendation effectiveness.

**Keywords:** text sentiment analysis; knowledge graph; graph attention networks; personalized recommendations; aspect of sentiment knowledge graph attention network (ASKAT); ABSA algorithm



**Citation:** Cui, Y.; Zhou, P.; Yu, H.; Sun, P.; Cao, H.; Yang, P. ASKAT: Aspect Sentiment Knowledge Graph Attention Network for Recommendation.

*Electronics* **2024**, *13*, 216. <https://doi.org/10.3390/electronics13010216>

Academic Editor: Dimitris Kanellopoulos

Received: 16 November 2023

Revised: 25 December 2023

Accepted: 29 December 2023

Published: 3 January 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Recommender systems are present in every aspect of our daily online lives due to their powerful ability to filter the ever-growing volume of data, enabling us to quickly access the information we all need in our fast-paced modern lives [1,2], such as product recommendations on the Amazon shopping site, music recommendations in NetEase Cloud Music, etc. Recommendation algorithms often use historical interaction data for modelling and learning to predict the probability of interaction between a user and a candidate item. Among the various recommendation techniques, collaborative filtering has achieved great success simply and effectively based on the idea that “things are grouped together, and people are grouped together” [3–5]. However, collaborative filtering methods generally suffer from data sparsity and cold-start problems when there are less interaction data or when new users join. At the same time, most of the methods focus only on user ratings, which are only the overall evaluation of an item by users, and it is difficult to infer real user preferences from them. In addition, the specific aspects that different users care about for

the same item may be different, even if they have the same ratings. Collaborative filtering is unable to analyse such fine-grained information and thus cannot accurately model user preferences, leading to biased recommendation results [6].

In order to improve the performance of recommender systems, researchers have tried to use various schemes to solve the above problems. These schemes are mainly review and knowledge-graph-based recommendation. In order to facilitate the reader's understanding, we categorised these recommendation algorithms, and each category included aspects such as the main technical features, representative models, and limitations, as shown in Table 1.

**Review-based recommendations.** Review data are readily available in various application platforms on the Internet. Textual reviews have more detailed user opinions and item attributes than ratings [7], and methods based on reviews and ratings effectively solve the data sparsity and cold-start problems. These methods use different strategies to extract features from reviews, and early research attempted to model topics from reviews, obtain user preferences, and then use collaborative filtering for recommendations [8–10]. Some other researchers have used clustering to process review texts to categorize users and items from reviews and thus improve the recommendation performance based on category similarity [11–13]. In recent years, deep-learning-based approaches have become popular, which use deep learning methods to learn user and item representations from reviews. They are mainly divided into two categories. One is a document-level recommendation, which splices and integrates user and item reviews into a single document for learning, and this type of method can obtain feature representations from a global perspective. Typical representative works include the DeepCoNN [14] model, which uses a deep learning approach exclusively utilizing two parallel networks to integrate the information of the reviews written by the user into two documents used to learn the representation of the user and the item. The D-Attn [15] model uses local attention to learn the user's preferences and the item's attributes and global attention to focus on the overall semantic information of the review text. Another category is single comment level recommendation; this type of approach models each comment individually and then aggregates these features to obtain a final representation of user and item features that are able to capture the user's preference for a specific item. Representative works include NARRE [16] and TARMF [17] models that consider the importance of different reviews to be different, proposing a review-level attention mechanism to calculate the weight of each review. The HUITA [18] model considers that not only the importance of different comments is different, but also the importance of different sentences and even different words in the same comment, so the representations of users and items are learned using three levels of attention at the word level, sentence level, and comment level, respectively. In addition, some new research combines machine learning with swarm intelligence approaches and it has proven to be able to achieve outstanding results in different areas [19–22]. Despite the progress made by these approaches, the modelling of embedded representations of users and items is still latent and does not accurately represent the personalized preferences of users and the personalized characteristics of items, which undoubtedly makes the recommendation performance suffer. There is also a sentiment-based approach [23–26]. This approach allows for fine-grained preference modelling of reviews to more accurately capture embedded representations of users and items, leading to further improvements in recommendation performance. In order to obtain a finer-grained sentiment analysis, researchers have conducted much work in recent years to identify the need for finer-grained aspect-level opinions and sentiments, known as aspect-based sentiment analysis (ABSA), which has received more and more attention. This approach is mainly implemented using ABSA techniques. ABSA is a field of research that analyses a variety of sentiment elements at the aspect level, and the main line of its research (a variety of specific tasks) is, that, given a text, the goal is to obtain one or several sentiments elements. ABSA is an essential fine-grained sentiment analysis technique that aims to analyse and understand human perspectives at the aspect level [27,28]. ABSA is an important technique in the field of natural language processing. However, it has been applied to review-based recommendation algorithms by more and more researchers be-

cause of its outstanding advantages in text processing [12,25,29]. For example, Li et al. [29] proposed a predictive model for user review ratings based on capsule networks. The model extracts viewpoints (viewpoints) and aspects from review documents, treating them as logical units. It designs an emotion capsule structure to reason about the representations of logical units and emotions, as well as user ratings prediction. Sung-Jun Park et al. [12] constructed an emotion-aware knowledge graph by analysing users' ratings and reviews of items, and they used a reinforcement learning strategy to make item recommendations and inferences. However, these approaches are overly dependent on the accuracy of external sentiment analysis tools.

**Knowledge graph-based recommendations.** The knowledge graph (KG) is a semantic network consisting of knowledge and relationships between knowledge, a large-scale knowledge base [30–33]. Knowledge graphs are rich in nodes and associations between nodes, which can be used to organise knowledge efficiently. Knowledge graphs have been gradually introduced into recommender systems due to their great success in various domains and have made good progress [34–38]. For example, RippleNet [36] proposed an end-to-end framework to iteratively propagate user preferences based on ratings using users' historical clicked items in the knowledge graph to expand users' potential interests. KGAT [34] addressed the problem of modelling each user interaction as an independent data instance in previous approaches by using users' historical interaction data to model higher-order relationships in a knowledge graph attention network, thereby extracting collaborative signals from collective behaviours. KGCL [35] addressed the sparsity and noise problem of KGs by designing a generalized knowledge graph comparative learning framework to mitigate the information noise of knowledge graph-enhanced recommender systems. These efforts use ratings data to propagate and mine latent user preferences in the knowledge graph; however, they cannot model personalised information about users and items at a fine-grained level.

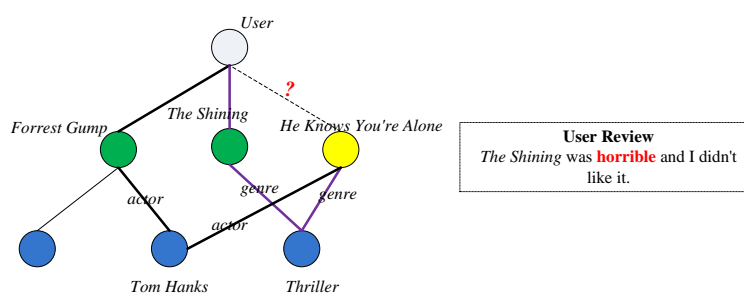
**Table 1.** Categories of recommendation models.

Categories		Main Technical Features	Algorithmic Models	Limitation
Recommendations based on reviews	Theme-based approach	Modelling the theme	[8–10]	Insufficient granularity of topics
	Clustering-based approach	Categorise users and items using clustering Recommendations based on category similarity.	[11–13]	Inability to model at a fine-grained level
	Deep learning-based approach (document level)	Integrate reviews as documents for learning	[14,15]	Modelled embedded representations are latent and do not accurately represent personalised preferences
	Deep learning- based approach (single comment level)	Modelling each review individually	[16–18]	
	Sentiment-based approach	Fine-grained preference modelling of reviews based on sentiment	[12,23–26,29]	Over-reliance on external sentiment analysis tools
Knowledge graph based recommendation		Using historical interactions to propagate user preferences in the knowledge graph	[34–39]	Inability to model personalised information at a granular level

In addition to this, these methods have the following problems: existing methods extract aspect items and sentiment information from reviews, but they cannot effectively

correlate this information during coding, resulting in the inability to fully utilize extracted features. On the other hand, the existing methods only focus on the aspects that users like, while ignoring the aspects that users dislike, and they cannot completely model users' real preferences. We believe that by effectively integrating the information extracted from reviews with knowledge graphs, the important information extracted from review texts can be more fully utilized, and the recommendation performance can be effectively improved. At the same time, focusing on user disliked aspects based on the aspect sentiment information extracted from reviews can more accurately model user preferences, thus improving the accuracy of recommendations.

To highlight our motivation, a detailed explanation is provided in Figure 1. A user likes the movie *Forrest Gump*, and from the reviews, it is known that the user is interested in the actor Tom Hanks. According to the idea of collaborative filtering, propagating the user's interest in the knowledge graph is likely to recommend Tom Hanks in another movie, *He Knows You're Alone*. However, from the review of the movie *The Shining* that the user has seen, it is known that the user does not like the thriller genre and even has some aversion to it. This is contradictory to the possible recommendation of the thriller genre movie *He Knows You're Alone*. So, it is very necessary to pay attention to the aspects that the user dislikes.



**Figure 1.** Example of the need to focus on aspects that the user dislikes. The previous approach, which propagated user interests based on their interaction history (green nodes), was likely to recommend yellow node movies (thriller genre). However, it was learnt from user comments that the user disliked thriller movies.

Based on the above analysis this, we propose the Aspect-based Sentiment Knowledge Graph Attention Network (ASKAT) model, which aims to provide users with more accurate and personalised recommendation results. ASKAT first processes the review text using a text summarization algorithm to remove noisy data and unimportant information from the text. This differs from existing approaches in that we then used the popular ABSA algorithm to extract aspectual items and the corresponding sentiment, without relying on the accuracy of external sentiment analysis tools. Up to this point, the fine-grained aspect items and sentiments of users' concerns were extracted from the reviews. In order to overcome the difficulty that the information extracted from reviews cannot be fully utilized, we effectively aligned and fused the features extracted from reviews with the knowledge graph. After that, we proposed a new graph attention network to aggregate neighbour information using a sentiment-aware attention mechanism. Meanwhile, in order to completely model real user personalized features, we designed a Deleting Negative Affective Nodes Strategy (DNANS) to focus on user-disliked aspects of user review features. Unlike existing work, our work effectively aligned and fused the important information extracted using ABSA techniques with the knowledge graph, so that this information could be more fully utilised. At the same time, our work not only focused on the aspects that users liked, but also paid more attention to the aspects that users disliked, so as to more accurately grasp the personalised needs of users. Our work jointly used user ratings and reviews data to uncover personalised user preferences and personalised features of items for personalised recommendations.

Finally, we experimented on three real scenario datasets, and the experimental results showed a significant improvement in the performance of our model relative to the state-of-the-art recommendation model. In summary, our main contributions are as follows:

- We applied text summarization techniques with ABSA to knowledge-graph-aware recommendation work.
- To solve the underutilization of review information, we effectively aligned and fused the review features with the knowledge graph.
- We proposed a new aggregation strategy to aggregate actual user-personalized features to achieve the goal of knowing what is good and what is bad.
- Experiments were conducted on three real datasets to demonstrate the effectiveness of ASKAT on several state-of-the-art baselines.

The rest of the paper is organized as follows. Section 2 defines some basic preparatory knowledge and notations. Section 3 details the implementation of the ASKAT model. Section 4 describes the dataset, the baseline model, and the results of the experiments. Section 5 contains our conclusions.

## 2. Theoretical Framework

In this section, we first introduce some basic knowledge and notation related to our proposed ASKAT model, followed by a formulaic treatment of the problem being studied.

**Definition 1.** *User-Item Interaction Graph.*

In the recommendation domain, users' interaction history is usually used on items to mine useful information [1]. Generally, we denote the set of users  $u$  by  $U$  and the set of items  $i$  by  $I$ , respectively. Here, we describe the user-item interaction graph as  $\mathcal{G}_1 = \{V, E\}$ , where  $V$  is the node of the graph denoting the concatenation of user  $U$  with item  $I$ .  $E$  is the edge of the graph, denoted as  $y_{ui} = 1$  if user  $u$  has an interaction with item  $i$ .

**Definition 2.** *Knowledge Graph.*

We use  $\mathcal{G}_2 = \{(h, r, t)\}$  to denote an item knowledge graph, e.g., book knowledge graph and movie knowledge graph [40]. KG contains a large number of entity-relationship-entity triples  $(h, r, t)$ .  $h$  and  $t$  are the head entity and the tail entity, respectively, which belong to  $\mathcal{E}$ .  $r$  is the relationship between the entities, which belongs to  $\mathcal{R}$ . For example, the triples in movie recommendation (*The Shawshank Redemption*, directed by, Frank Darabont), and in book recommendations (*Les Misérables*, author, Victor Hugo).

**Definition 3.** *Collaborative Knowledge Graph.*

In this paper, we merge the bipartite graph (which represents user-item interactions) and the item knowledge graph into a collaborative knowledge graph (CKE) [34]. First, we represent the bipartite graph in the form of a triple  $(u, \text{interact}, i)$ . Then, the two-part graph in the form of triples is merged with the knowledge graph to form CKE, which is denoted as  $\mathcal{G} = \mathcal{G}_1 \cup \mathcal{G}_2 = \{(h, r, t) | h, t \in \mathcal{E}', r \in \mathcal{R}'\}$ ,  $\mathcal{E}' = \mathcal{E} \cup U$ ,  $\mathcal{R}' = \mathcal{R} \cup \{\text{Interact}\}$ .

**Definition 4.** *User Reviews.*

Our focus is on extracting fine-grained user preference information from reviews. The initial format of the reviews we used was dictionary data in json format  $\{\text{overall}, \text{reviewTime}, \text{reviewerID}, \text{item}, \text{reviews}\}$ . Where *overall* is the rating, *reviewTime* is the time of the review, *reviewerID* is the ID of the reviewer, *item* is the item of the review, and *reviews* is the specific text of the review. For ease of description and usage, we simplified and denoted the reviews as  $\mathbb{R} = \{\gamma_{ui} | u \in U, i \in I\}$ . The symbol  $\gamma_{ui}$  denotes the review that user  $u$  has allocated to item  $i$ .



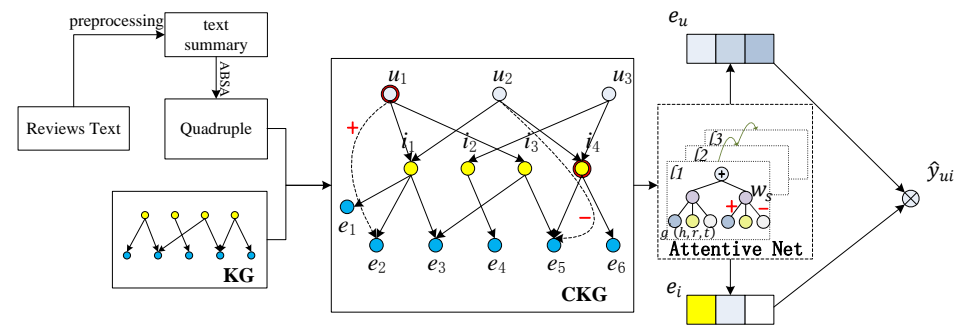
### Definition 5. Task Formulation.

Inputs: user  $u$ , candidate item  $i$ , collaborative knowledge graph  $\mathcal{G}$ , and reviews  $\mathbb{R}$ . The reviews contain two aspects, the user's historical reviews on the one hand and the items' historical reviews on the other.

Output: the predicted probability  $\hat{y}_{ui}$  that user  $u$  clicks on item  $i$ .

### 3. The Proposed Model

In this section, we introduce the proposed ASKAT model. The general architecture of ASKAT is shown in Figure 2. ASKAT is used to extract user's aspectual sentiment from reviews and combine it with the knowledge graph to improve the performance of the recommender system. ASKAT takes as inputs the user  $u$ , the candidate item  $i$ , the collaborative knowledge graph  $\mathcal{G}$ , and the review  $\mathbb{R}$ , and outputs the predicted probability  $\hat{y}_{ui}$  that the user  $u$  clicks on the item  $i$ . The user's aspectual sentiments are then analysed using the ABSA algorithm. ASKAT first uses the ABSA algorithm to perform aspect-sentiment analysis on the reviews, from which aspect items and sentiments of interest to the user are extracted. Subsequently, an aspect-sentiment-enhanced collaborative knowledge graph is achieved by aligning the extracted aspect items with the collaborative knowledge graph. Finally, in order to more accurately obtain the complete user preferences and focus on the aspects that the user dislikes, a true personalised preference-aware graph attention network is designed to capture the user and item representations. The details of the model will be elaborated upon as follows.



**Figure 2.** ASKAT model framework diagram. The top leftmost side of the figure is the processing flow for reviews, containing text summary and ABSA, and the bottom side is the original knowledge graph. The CKG in the middle is the collaborative knowledge graph for aspect sentiment enhancement, where “+” indicates positive sentiment and “−” indicates negative sentiment. On the far right is the personalised preference perception graph attention network.

#### 3.1. Aspect-Based Sentiment Analysis for Reviews

In this section, aspect-based sentiment analysis of reviews is described in detail. Before aspect sentiment extraction, the reviews text is first preprocessed using text summarization techniques, and then aspect items and the corresponding sentiment are extracted from the reviews using aspect-based sentiment analysis techniques.

##### 3.1.1. Text Summarization

Users' review data are usually disorganized and may be tedious and diverse. Undoubtedly, the noise in the reviews will affect the extraction results of aspect terms and sentiment polarity. Therefore, before extracting the aspect terms, etc., from the reviews, the data should be preprocessed first to improve the accuracy and efficiency of the extraction. Text summarization is a kind of extracting, summarizing, or refining the key information of the text or the collection of text through various techniques to summarize and display the main content or the general idea of the original text [41]. Text summarization technology is one of the key technologies to improve the efficiency of people's access

to effective information in the era of information explosion. How to distil key information from redundant, unstructured long text to form a concise and smooth summary is the core problem of text summarization [42]. In short, text summarization is an information compression technique. In view of this, this paper uses text summarization techniques to refine and streamline unstructured review texts. Among the classical text summarisation techniques SWAP-NET [43] directly uses the Seq2Seq model to alternately generate index sequences of words and sentences for the extractive summarisation task. However, this model scores and selects separately and cannot take advantage of the relationship between sentences. PGNet [44] is a Seq2Seq model based on the attention mechanism with the addition of copy and coverage mechanisms. The literature [45] uses the BERT model for text embedding and KMeans clustering to identify sentences close to the centroid in order to select summaries. We compared these algorithms and finally chose to use the method in [45] for comment text preprocessing. The extraction of comment text summaries is performed using BERT. The advantages over other models are two-fold, one is that the model can be customised to extract the number of sentences as required, and the other and most important is that the method uses a pre-trained model for text summarisation, which has better generalisation and representation capabilities.

### 3.1.2. The Aspect and Sentiment Extraction

Reviews on e-commerce platforms contain many users' sentiments and opinions about items. Analysing and mining user preferences from them can help to improve products and services and conduct better business activities [46]. The analysis of reviews using ABSA provides new perspectives to improve the accuracy and personalization aspects of recommender systems. In this paper, we use the mature and advanced ABSA algorithm [47] to perform sentiment analysis on reviews to obtain fine-grained user personalized preferences and apply them to recommender systems.

We used the algorithm integrated into the Pyabsa framework [48] to perform aspect-based sentiment analysis on preprocessed review text to extract aspect items and sentiment polarity. The input of the algorithm was reviews and the output was the tuple  $\{(a, s) | a \in A, s \in S\}$ , where  $a$  denotes the aspect term associated with an item's features,  $A$  is the set of aspect terms,  $s$  denotes the sentiment polarity, and  $S$  is the range of sentiment polarity  $\{positive, neutral, negative\}$ . To facilitate the distinction, we extracted the aspect terms and sentiment polarity by adding the user and item corresponding to the review, and denoted the extracted information as  $\{(u, i, a, s) | u \in U, i \in I, a \in A, s \in S\}$ , where  $u$  denotes the user and  $i$  denotes the item. This quaternion represents the sentiment polarity of the aspect item  $a$  of item  $i$  extracted from the user's reviews of item  $i$  for that user as  $s$ , indicating the user's fine-grained aspectual sentiment.

### 3.2. Aspect-Sentiment Enhanced Collaborative Knowledge Graph

The collaborative knowledge graph contains the user's interaction history and the item's attribute knowledge. Moreover, the aspect sentiment information extracted from reviews contains rich user personalized preferences. Because of the different review styles and linguistic expressions of users, the same referring aspect item in the review texts of different users varies greatly and is ambiguous. For example, user A and user B both commented on *Tim Robbins*, the star of the movie *The Shawshank Redemption*, but in A's description, he is referred to as "*Tim*", while in B's description, it is "*Robbins*", which actually This refers to the same node "*Tim Robbins*" in the knowledge graph. In addition, it should be noted that the knowledge in the knowledge graph is not in the same space as the knowledge in the text reviews, so how do we link the different referents to the corresponding nodes in the knowledge graph? This is the scope of entity alignment in the knowledge graph. After researching this paper, we finally chose to use the word2vec approach to align aspect items with the attribute nodes of items in the knowledge graph. Google's pre-trained word vector model [49] is a high-quality representation of word vectors learned from a dataset containing 1.6 billion words. The vector representations of aspect items

and item attribute nodes are obtained by using this model to map them into vectors in the same space in an unsupervised manner quickly and easily. Then, the similarity between them is calculated by similarity, and they are sorted according to the similarity scores to finally realize the alignment. After alignment, aspect sentiment information can be linked to the collaborative knowledge graph to form a collaborative knowledge graph for aspect sentiment enhancement. We denote this linkage as  $\{(u, i, e, s) | u \in U, i \in I, e \in \mathcal{E}', s \in S\}$ , where  $e$  denotes the attribute node connected to the item entity in the CKG, i.e., the user  $u$  holds the sentiment  $s$  for the aspect  $e$  of item  $i$ .

### 3.3. Truly Personalized Preference-Aware Graph Attention Networks

Our model is constructed on the basis of the knowledge graph, and the number of neighbour nodes of each node in the knowledge graph may not be the same, which are unstructured data. Using graph neural networks to process graph data has a natural advantage over many deep learning methods [50], so we used graph attention networks from graph neural networks to construct the model in this paper. In this section, we detail our proposed truly personalised preference-aware graph attention network. In the following section, we address four aspects: embedding layer, propagation layer, prediction layer, and model optimisation.

#### 3.3.1. Embedding Layer

We used TransR [51] for the embedded representation of knowledge graphs to vectorize entities and relationships in collaborative knowledge graphs, which has the advantage of effectively preserving the graph structure.

#### 3.3.2. Propagation Layers

In this section, we describe in detail the propagation of users' preferences in the graph. We always believe that true personalization can only be achieved by knowing both the good and bad aspects of users' interests and preferences. This statement can be interpreted to mean that when aggregating user preferences, one cannot only focus on the aspects that the user likes and favours, but must also consider the aspects that the user dislikes. The personalized representation of the user obtained in this way is truly comprehensive and complete. We specially designed a Delete Negative Affective Node Strategy (DNANS), which deleted a node when aggregating neighbouring nodes if the user's sentiment towards the node was negative. The recommendation obtained according to such personalized preferences was more accurate, and most importantly, it was in line with the user's taste preferences so as to achieve truly personalized recommendations. For this purpose, we specially designed an attention weight function as follows:

$$w_s = \begin{cases} 0 & , \text{if } s(u, i, e) = -1 \\ \alpha & , \text{if } s(u, i, e) = 1 \\ 1 - \alpha & , \text{other} \end{cases} \quad (1)$$

where  $\alpha$  is a sentiment coefficient hyperparameter that takes values between 0 and 1.  $s(u, i, e)$  denotes the sentiment of user  $u$  towards aspect  $e$  of item  $i$ . The weight of node  $e$  scores  $\alpha$  when  $s$  is a positive sentiment, and the weight of node  $e$  scores  $1 - \alpha$  when  $s$  is a neutral sentiment, and, in particular, note that when  $s$  is a negative sentiment, we discard the aggregation of the node, and set the node's weight to zero. We recursively propagate the embedding on the architecture of graph convolutional networks [52,53], using the idea of graph attention networks [54], describing a single layer and generalizing to multiple layers.

In a knowledge graph, entities are connected to each other through relationships, where an entity can be connected to multiple neighboring entities, and to other nodes over multiple hops through higher-order connected entities. In this way, a user entity can be connected to all interaction history items, which in turn are connected to their respective attribute feature nodes. Items can enrich their own feature representations when aggregating attribute information and then contribute to the user. In this way, the attribute



information of the user's interaction history items can be propagated to the user. We build on this idea to propagate information between entities on the knowledge graph.

For entity  $h$ , the set of all its triples can be denoted as  $N_h = \{(h, r, t) | (h, r, t) \in \mathcal{G}\}$  and its neighbors can be denoted as follows:

$$e_{N_h} = \sum_{(h,r,t) \in N_h} g(h, r, t) e_t \quad (2)$$

where  $e_t$  denotes the embedding representation of the tail entity  $t$  connected to the neighbouring head entity  $h$ .  $g(h, r, t)$  controls the weight factor of the information propagated from entity  $t$  to entity  $h$  in the triad  $(h, r, t)$ . We implement  $g(h, r, t)$  using a combination of sentiment-aware attention and relational attention, which is formulated as follows:

$$g(h, r, t) = g'(h, r, t) + w_s \quad (3)$$

where  $w_s$  denotes the sentiment perception weight extracted from the reviews, the first term in Equation (3) denotes the relational attention weight  $g'(h, r, t) = (e_t^r)^T \tanh(e_h^r + e_r)$  (the score depends on the distance between entities  $h$  and  $t$ ),  $e_t^r$  is the projective representation of  $e_t$  in the relational  $r$ -space, and  $e_h^r$  is the projection of the embedding representation  $e_h$  of the head entity  $h$  in relational  $r$ -space. In this paper, we use the softmax function to normalize the coefficients of all neighbouring nodes of entity  $h$ :

$$g(h, r, t) = \frac{\exp(g'(h, r, t))}{\sum_{(h,r,t) \in N_h} \exp(g'(h, r, t))} \quad (4)$$

Ultimately, the neighbour representation  $e_{N_h}$  of entity  $h$  can be obtained by biased aggregation of the attention scores, which incorporates both the personalized preference information in the reviews and the collaborative information freely captured in the graph. These data can be used as support for exploring interpretability.

For a single layer, the new representation of entity  $h$  can be obtained by aggregating its own embedded representation  $e_h$  and its neighbour representation  $e_{N_h}$ , with the following formula:

$$e_h' = f(e_h, e_{N_h}) \quad (5)$$

As in the literature [34], we also used the three aggregation methods, GCN, GraphSage, and Bi-Interaction, to obtain  $e_h'$ ; we will not elaborate here.

### 3.3.3. Prediction Layer

Above is the process of single-layer aggregation, and we stacked more propagation layers to aggregate higher-order neighbourhood information by recursion.

After  $L$  layers of aggregation, the final representations of all nodes in the knowledge graph were finally obtained, from which the embedding representation  $e_u$  of user  $u$  and the embedding representation  $e_i$  of candidate item  $i$  were taken out, respectively, and the predicted probability  $\hat{y}_{ui}$  of this user clicking on item  $i$  was obtained by the inner product:

$$\hat{y}_{ui} = e_u e_i \quad (6)$$

### 3.3.4. Model Optimization

To optimize our model, we computed the knowledge graph embedding loss and the recommendation algorithm loss, which exploits the widespread use of Bayesian Personalized Ranking (BPR) loss with the loss function defined as follows:

$$\mathcal{L} = - \sum_{(u,i,j) \in \mathcal{O}} \ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \mathcal{L}_{KG} + \lambda \|\mathcal{F}\|_2^2 \quad (7)$$

The first of these is the BPR loss, the second is the knowledge graph embedding loss, and the last is the L2 regularization equation to prevent model overfitting.

## 4. Experiments

In this section, we evaluate our proposed method in two scenarios: click-through prediction and top-k recommendation.

### 4.1. Datasets

We used Book, Movie, and Yelp datasets that have review information in our experiments. The Book dataset uses Amazon book, which is a large-scale book review dataset provided by Amazon.com. The Movie dataset is MovieLens, which is the most classic dataset in the field of recommender systems and contains user ratings of movies. The Yelp dataset is the dataset provided by the largest review website in the US, which contains merchant, review, and user data. As the MovieLens (<https://grouplens.org/datasets/movielens/> (accessed on 8 July 2023)) dataset, which is commonly used in the recommendation domain, does not have corresponding review data, we re-crawled the data with reviews on the IMDB (<https://www.imdb.com> (accessed on 8 July 2023)) website based on the IMDB numbers of the movies in the original dataset and created a new Movie dataset. The Amazon book (<http://jmcauley.ucsd.edu/data/amazon> (accessed on 8 July 2023)) dataset was downloaded from [55,56]. The Yelp (<https://www.yelp.com/dataset> (accessed on 8 July 2023)) dataset consisted of data from 11 metropolitan areas with about 150,000 merchants, 6.99 million reviews, and 200,000 image data. In addition to the dataset with reviews and ratings, we also used knowledge graph data. The knowledge graph data for the Movie dataset were from [39], and the knowledge graph data for the Amazon book dataset were from [34]. The knowledge graph for the Yelp dataset was constructed by extracting the knowledge of the items using our local business information network. The statistical information of the dataset is shown in Table 2.

**Table 2.** Detailed statistical tables for the three datasets. The table contains the number of users, items, and reviews for each dataset, as well as information on the dataset's densities and the knowledge graphs used (number of entities, relationships and triples).

	Movie	Amazon-Book	Yelp
users	23,641	14,762	42,464
items	23,362	24,915	150,337
reviews	752,782	311,887	1,746,230
density	0.136%	0.085%	0.027%
entities	102,569	113,487	155,466
relations	32	39	41
KG triples	499,474	2,557,746	1,566,773

### 4.2. Baselines

To evaluate the models, we selected state-of-the-art baseline models for comparison. These methods were as follows:

**KGAT [34]:** In response to the problem of modelling each user interaction as an independent data instance in previous approaches, the historical user interaction data are used to model higher-order relationships in a knowledge graph attention network, thus extracting collaborative signals from collective behaviours.

**CFKG [57]:** This is an interpretable recommendation algorithm based on knowledge embedding. A knowledge-based representation learning framework is first used to embed the knowledge base, and then a soft matching algorithm is proposed on this basis to generate personalized explanations of recommended items.

**NFM [58]:** This model addresses the shortcomings of FM [59] that cannot cope with real data with a complex structure, and proposes a strategy of FM fusion into DNN to make the two perfectly articulate, which can model higher-order feature interactions.

LightGCN [60]: LightGCN is a graphical convolutional neural network applied to the recommendation system. It abandons the feature transformation and nonlinear activation commonly used in GCN, and does not use self-informative links.

KGCL [35]: To address the sparsity and noise problems of KG, this model proposes a knowledge-graph-based contrast learning framework to mitigate the information noise in the recommendation modelling process and learn user preferences more accurately.

RippleNet [8]: This model is the first to integrate embedding-based and path-based approaches into KG-aware recommender systems. The user's preference propagation is analogized to ripple propagation in KG.

KGCN [39]: This is the classic knowledge graph and GNN-based approach. It uses GCN to automatically capture the higher-order structural and semantic information of items in the knowledge graph and explores the user's personalized preferences by learning the importance of relationships with the user.

#### 4.3. Experimental Settings

We implemented the ASKAT model using Tensorflow 1.15.0 and Python 3.8. The embedding sizes of the models were all set to 64 due to computational cost. We used the default Xavier initializer to handle the initial parameters of the models, and the optimizer used Adam. We divided each dataset into three subsets: training set (60% of the original dataset), validation set (20%), and test set (20%). We trained the models in two phases alternately, namely, recommendation model training and knowledge graph embedding training. We set the number of layers of ASKAT to 3, and the output dimension size of each layer was set to 64, 32, and 16, respectively, and we used LeakyReLU for the activation function of the last layer. Meanwhile, we set up an early-stop mechanism in our experiments, which stopped the training early when there was no improvement in the performance for 50 consecutive epochs on the validation set. For each layer of the model, we used the GCN aggregator. We also performed a grid search on the hyperparameters: the learning rate was tuned between {0.01, 0.005, 0.001, 0.0005, 0.0001, and 0.00005}. In addition, we set the sentiment coefficient hyperparameter to be adjusted between 0 and 1. We used recall and ndcg to evaluate CTR predictions and precision@k to evaluate top-k recommendations.

#### 4.4. Results

##### 4.4.1. Comparison Experiment

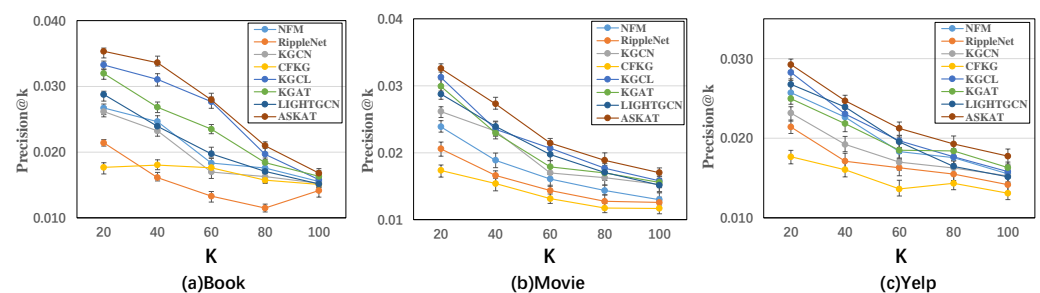
The results of our comparison experiments in two recommendation scenarios, CTR prediction and top-k recommendation, are shown in Table 3 and Figure 3. Table 3 counts the recall and ndcg of our proposed model ASKAT and the other seven methods. Figure 3 shows the precisions of the eight methods in the top-k recommendation, counting the experimental results of top-20, top-40, top-60, top-80, and top-100, respectively. By analysing the experimental results, in general, we achieved the following observations:

- From Table 3 and Figure 3, we can conclude that ASKAT consistently outperformed all baseline models;
- From Table 2, we can learn that as far as sparsity was concerned, the Yelp dataset had the highest sparsity, Book was second, and Movie was relatively denser. From Table 3, the overall experimental results on the three datasets showed that the average performance improvement on the Yelp dataset was the most significant, while the denser Movie dataset had the least performance improvement. From this, it can be judged that our model effectively alleviated the data sparsity problem of recommender systems;
- We also found that not all knowledge-graph-based methods outperformed traditional methods, indicating that the effective utilization of knowledge graph information in recommendation is crucial, or else the model performance will instead be affected by introducing too much noise;
- From Table 3, we also observed that GCN-based models such as KGCN and KGAT performed significantly better than other KG-based methods, which indicates that the ability of GCN in processing graph data should not be underestimated;

- The performance of ASKAT on the Movie and Book datasets was significantly higher than the Yelp dataset. We analysed the possible reason for this as that the reviews on the Movie and Book datasets were more focused on a single domain, such as movies or books. On the contrary, reviews on the Yelp dataset were more dispersed as they related to a wide range of domains such as restaurants, shopping centres, hotels, and travel. The model handled single domains better when learning features from reviews, while adapting to multi-domain scenarios was limited.

**Table 3.** Comparison of the overall performance of CTR prediction. The numbers in bold indicate that the improvement in our model over all baselines was statistically significant with  $p < 0.05$  under  $t$ -test.

Model	Movie		Amazon-Book		Yelp	
	Recall	ndcg	Recall	ndcg	Recall	ndcg
NFM	0.1490	0.1390	0.1678	0.1551	0.0710	0.1314
RippleNet	0.1414	0.1357	0.1541	0.1346	0.0614	0.1322
KGCN	0.1536	0.1377	0.1615	0.1473	0.0703	0.1055
CFKG	0.1447	0.1216	0.1358	0.1425	0.0570	0.1144
KGCL	0.1507	0.1417	0.1805	0.1674	0.0806	0.1416
KGAT	0.1580	0.1406	0.1785	0.1701	0.0762	0.1367
LightGCN	0.1532	0.1367	0.1719	0.1361	0.0783	0.1259
ASKAT	<b>0.1636</b>	<b>0.1465</b>	<b>0.1852</b>	<b>0.1736</b>	<b>0.0841</b>	<b>0.1465</b>



**Figure 3.** Experimental results of precision@k metrics for three datasets in the Top-k recommendation scenario. The figure shows the experimental results (precision@k metrics) of our ASKAT model and other comparative models on the three datasets of Book, Movie, and Yelp, respectively, with the inclusion of error bars at each point. The horizontal coordinate indicates the value of K and the vertical coordinate indicates the corresponding precision value.

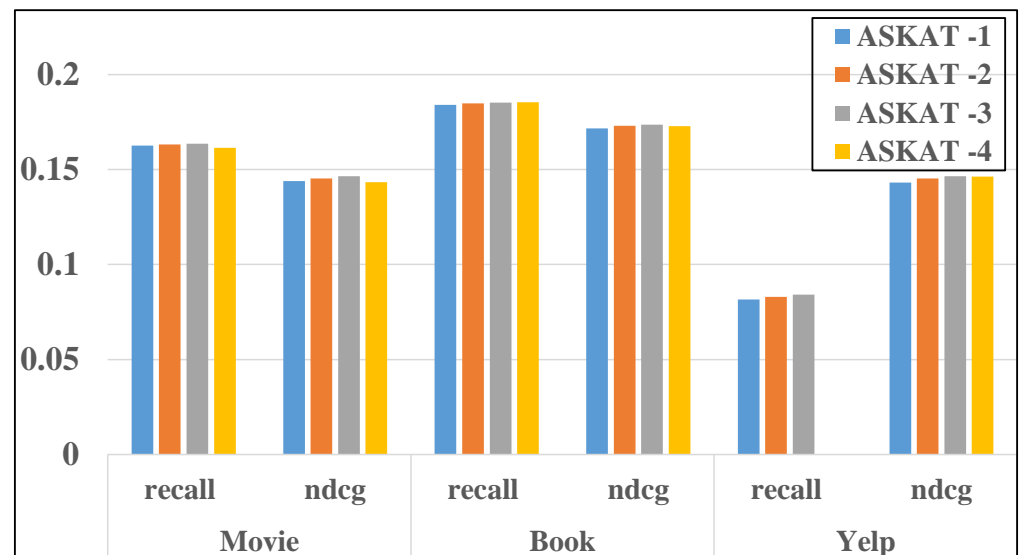
#### 4.4.2. Ablation Experiment

We performed ablation experiments on three datasets to analyse the effect of the number of embedded propagation layers, different aggregators, and several different variants of the model on the model, respectively. We evaluated the experimental results of the ablation experiments in terms of both recall and ndcg as follows:

**Effect of the number of embedded propagation layers.** In Table 4, we analyse the effect of different number of layers on ASKAT. From the table, we can observe that the performance of the model basically showed an upward trend with an increase in the number of layers of embedding propagation, i.e., the more layers, the better the model. However, when the number of layers increased to a certain degree, the performance of the model began to show a decreasing trend. This indicates that when the number of layers increased, the utility of the higher-order features in the knowledge graph was compared with the introduced noise, and, obviously, the noise prevailed. According to the experimental results, the performance of the model was best when the number of layers was 3. In order to better present the results, we also visualised the results of this experiment, as shown in Figure 4.

**Table 4.** Effect of the number of propagation layers on the model.

Model	Movie		Amazon-Book		Yelp	
	Recall	ndcg	Recall	ndcg	Recall	ndcg
ASKAT-1	0.1626	0.1439	0.1842	0.1717	0.0816	0.1432
ASKAT-2	0.1632	0.1453	0.1848	0.1730	0.0830	0.1453
ASKAT-3	<b>0.1636</b>	<b>0.1465</b>	0.1852	<b>0.1736</b>	<b>0.0841</b>	<b>0.1465</b>
ASKAT-4	0.1615	0.1434	<b>0.1854</b>	0.1728	0.8128	0.1463

**Figure 4.** Impact of the number of propagation layers on the model. The figure shows the effect of different numbers of propagation layers on the ASKAT model, evaluating the results on two metrics: recall and ndcg. ASKAT-K denotes a variant of the ASKAT model with different numbers of layers (from 1 to 4), the vertical coordinate denotes the recall or the ndcg value, and the horizontal coordinate denotes the recall and the ndcg for each of the three datasets.

**Effects of different aggregators.** In this section, we analyse the effect of different aggregators on our ASKAT model. Our model was tested with three aggregators, GCN, GraphSage, and Bi-Interaction, respectively. The experimental results of the models using these three aggregators are shown in Table 5. Based on the experimental results, we observed that the performance was optimal when the model used the GCN aggregator. This may be related to the fact that we incorporated review sentiment features, and the GCN performance could perform better when incorporating review information aligned with the knowledge graph. This further illustrates the effectiveness of our approach based on review sentiment analysis. Our model effectively correlated the features extracted from reviews with the knowledge graph, which reduced the effect of noise to more accurately aggregate user preference information.

**Table 5.** Impact of the three aggregators on the model.

Model	Movie		Amazon-Book		Yelp	
	Recall	ndcg	Recall	ndcg	Recall	ndcg
GCN	<b>0.1636</b>	<b>0.1465</b>	<b>0.1852</b>	<b>0.1736</b>	<b>0.0841</b>	<b>0.1465</b>
GraphSage	0.1564	0.1457	0.1768	0.1654	0.0837	0.1460
Bi-Interaction	0.1534	0.1448	0.1722	0.1627	0.8245	0.1443



**Impact of ASKAT model variants.** In this subsection, we compare the performance of two variants of the ASKAT model against ASKAT in order to analyse the importance of some components of the model. The first variant is that the model did not use text summarization techniques for comment compression. The second variant is that the model did not use the delete negative affect node strategy (DNANS). From the experimental results in Table 6, it can be seen that the two components of the text summarization step and the DNANS were very necessary for the model. Especially when the DNANS failed, the performance degradation of the model was relatively obvious. This shows that the idea of “knowing what is good and what is bad” is crucial.

**Table 6.** Comparison of ASKAT and its variants.

Model	Movie		Amazon-Book		Yelp	
	Recall	ndcg	Recall	ndcg	Recall	ndcg
ASKAT w/o textS	0.1624	0.1448	0.1840	0.1715	0.0838	0.1448
ASKAT w/o DNN	0.1614	0.1425	0.1821	0.1706	0.0817	0.1430
ASKAT	<b>0.1636</b>	<b>0.1465</b>	<b>0.1852</b>	<b>0.1736</b>	<b>0.0841</b>	<b>0.1465</b>

## 5. Conclusions

In this paper, we propose the ASKAT model. It is a framework that utilizes advanced techniques in the field of natural language processing (NLP) [45,47,48] to perform sentiment-aware feature extraction of user reviews and thus apply it to KG-based recommendation algorithms. It effectively fuses the user’s aspectual sentiment in the reviews with the collaborative knowledge graph [40] and introduces it into the recommender system. Our ASKAT model uses an advanced ABSA algorithm [47] to extract aspectual sentiment, overcoming the over-reliance on the accuracy of external sentiment analysis tools. Review features are fully utilized through effective fusion with the knowledge graph. Meanwhile, the use of the remove negative nodes strategy focuses on both the aspects that users like and dislike, which can completely model users’ real preferences. We conducted extensive experiments on three real datasets [34,39,55,56], and the results show that ASKAT has significant advantages over strong baseline models. By analysing the experimental results, we believe that there are three reasons our model achieved better results: firstly, we used text summarisation techniques and the ABSA algorithm to extract aspectual sentiment from the reviews; secondly, we effectively fused the features extracted from the reviews with collaborative knowledge graphs; thirdly, our proposed DNANS strategy could fully model user preferences to achieve truly personalised recommendations (see Section 4.4 for a detailed analysis of the experimental results). The proposed model can help practitioners to provide more accurate personalised recommendation services, and also provide new ideas for academic researchers to mine personalised user preferences for recommendation systems. In addition, our model has some limitations, such as not being able to obtain more personalised information when there are fewer user comments.

For future work, we intend to continue to investigate the effective utilization of review aspect sentiment in recommender systems. Using large language models (LLMs) for aspect sentiment analysis to process reviews is a promising research direction [61]. Also, it will be further explored for the effective fusion of multiple modal data to be applied to recommender systems. Further, with the rapid development of technology, privacy protection and bias of users in recommender systems are becoming more and more important. Therefore, we will increase the consideration of privacy and bias in our future work.

**Author Contributions:** Methodology, Y.C. and P.S.; software, Y.C. and P.Z.; writing—original draft, Y.C. and P.Z.; data collection and revision, Y.C., H.Y. and P.Y.; writing—review and editing, P.Y. and H.C.; Funding acquisition, P.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by the National Natural Science Foundation of China under Grant 61866031.

**Data Availability Statement:** All data are open public data and able to be downloaded free of charge.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

- Lu, J.; Wu, D.; Mao, M.; Wang, W.; Zhang, G. Recommender system application developments: A survey. *Decis. Support Syst.* **2015**, *74*, 12–32. [\[CrossRef\]](#)
- Gao, C.; Zheng, Y.; Li, N.; Li, Y.; Qin, Y.; Piao, J.; Quan, Y.; Chang, J.; Jin, D.; He, X.; et al. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Trans. Recomm. Syst.* **2023**, *1*, 1–51. [\[CrossRef\]](#)
- Hu, Y.; Koren, Y.; Volinsky, C. Collaborative filtering for implicit feedback datasets. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; pp. 263–272.
- Koren, Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 426–434.
- Wang, X.; He, X.; Wang, M.; Feng, F.; Chua, T.S. Neural graph collaborative filtering. In Proceedings of the 42nd international ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 165–174.
- He, X.; Chen, T.; Kan, M.Y.; Chen, X. Trirank: Review-aware explainable recommendation by modeling aspects. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, Melbourne, Australia, 18–23 October 2015; pp. 1661–1670.
- Shuai, J.; Zhang, K.; Wu, L.; Sun, P.; Hong, R.; Wang, M.; Li, Y. 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, Madrid, Spain, 11–15 July 2022; pp. 1283–1293.
- Xu, J.; Zheng, X.; Ding, W. Personalized recommendation based on reviews and ratings alleviating the sparsity problem of collaborative filtering. In Proceedings of the 2012 IEEE Ninth International Conference on e-Business Engineering, Hangzhou, China, 9–11 September 2012; pp. 9–16.
- Musat, C.C.; Liang, Y.; Faltings, B. Recommendation using textual opinions. In Proceedings of the IJCAI International Joint Conference on Artificial Intelligence, Beijing, China, 3–9 August 2013; pp. 2684–2690.
- Bao, Y.; Fang, H.; Zhang, J. Topicmf: Simultaneously exploiting ratings and reviews for recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, Quebec City, QC, Canada, 27–31 July 2014; Volume 28.
- Chen, L.; Wang, F. Preference-based clustering reviews for augmenting e-commerce recommendation. *Knowl.-Based Syst.* **2013**, *50*, 44–59. [\[CrossRef\]](#)
- Wang, J.; Zhao, W.; He, Y.; Li, X. Leveraging product adopter information from online reviews for product recommendation. In Proceedings of the International AAAI Conference on Web and Social Media, Oxford, UK, 26–29 May 2015; Volume 9, pp. 464–472.
- Sohail, S.S.; Siddiqui, J.; Ali, R. Feature extraction and analysis of online reviews for the recommendation of books using opinion mining technique. *Perspect. Sci.* **2016**, *8*, 754–756. [\[CrossRef\]](#)
- Zheng, L.; Noroozi, V.; Yu, P.S. 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, Cambridge, UK, 6–10 February 2017; pp. 425–434.
- Seo, S.; Huang, J.; Yang, H.; Liu, Y. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In Proceedings of the Eleventh ACM Conference on Recommender Systems, Como, Italy, 27–31 August 2017; pp. 297–305.
- Chen, C.; Zhang, M.; Liu, Y.; Ma, S. Neural attentional rating regression with review-level explanations. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 1583–1592.
- Lu, Y.; Dong, R.; Smyth, B. Coevolutionary recommendation model: Mutual learning between ratings and reviews. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 773–782.
- Wu, C.; Wu, F.; Liu, J.; Huang, Y. Hierarchical user and item representation with three-tier attention for recommendation. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 3–5 June 2019; pp. 1818–1826.
- Bacanin, N.; Stoean, R.; Zivkovic, M.; Petrovic, A.; Rashid, T.A.; Bezdan, T. Performance of a novel chaotic firefly algorithm with enhanced exploration for tackling global optimization problems: Application for dropout regularization. *Mathematics* **2021**, *9*, 2705. [\[CrossRef\]](#)
- Malakar, S.; Ghosh, M.; Bhowmik, S.; Sarkar, R.; Nasipuri, M. A GA based hierarchical feature selection approach for handwritten word recognition. *Neural Comput. Appl.* **2020**, *32*, 2533–2552. [\[CrossRef\]](#)
- Bacanin, N.; Zivkovic, M.; Al-Turjman, F.; Venkatachalam, K.; Trojovský, P.; Strumberger, I.; Bezdan, T. Hybridized sine cosine algorithm with convolutional neural networks dropout regularization application. *Sci. Rep.* **2022**, *12*, 6302. [\[CrossRef\]](#)
- Zivkovic, M.; Bacanin, N.; Antonijevic, M.; Nikolic, B.; Kvascev, G.; Marjanovic, M.; Savanovic, N. Hybrid CNN and XGBoost model tuned by modified arithmetic optimization algorithm for COVID-19 early diagnostics from X-ray images. *Electronics* **2022**, *11*, 3798. [\[CrossRef\]](#)

23. Zhang, Y. Incorporating phrase-level sentiment analysis on textual reviews for personalized recommendation. In Proceedings of the Eighth ACM International Conference on Web Search And Data Mining, Shanghai, China, 2–6 February 2015; pp. 435–440.
24. Pradhan, R.; Khandelwal, V.; Chaturvedi, A.; Sharma, D.K. Recommendation system using lexicon based sentimental analysis with collaborative filtering. In Proceedings of the 2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), Mathura, India, 28–29 February 2020; pp. 129–132.
25. Huang, C.; Jiang, W.; Wu, J.; Wang, G. Personalized review recommendation based on users' aspect sentiment. *ACM Trans. Internet Technol. (TOIT)* **2020**, *20*, 1–26. [\[CrossRef\]](#)
26. Park, S.J.; Chae, D.K.; Bae, H.K.; Park, S.; Kim, S.W. Reinforcement learning over sentiment-augmented knowledge graphs towards accurate and explainable recommendation. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event, 21–25 February 2022; pp. 784–793.
27. Do, H.H.; Prasad, P.W.; Maag, A.; Alsadoon, A. Deep learning for aspect-based sentiment analysis: A comparative review. *Expert Syst. Appl.* **2019**, *118*, 272–299. [\[CrossRef\]](#)
28. Peng, H.; Xu, L.; Bing, L.; Huang, F.; Lu, W.; Si, L. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 8600–8607.
29. Li, C.; Quan, C.; Peng, L.; Qi, Y.; Deng, Y.; Wu, L. 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, Paris, France, 21–25 July 2019; pp. 275–284.
30. Hogan, A.; Blomqvist, E.; Cochez, M.; d'Amato, C.; Melo, G.D.; Gutierrez, C.; Krrane, S.; Gayo, J.E.L.; Navigli, R.; Neumaier, S.; et al. Knowledge graphs. *ACM Comput. Surv. (Csur)* **2021**, *54*, 1–37. [\[CrossRef\]](#)
31. Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; Philip, S.Y. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 494–514. [\[CrossRef\]](#) [\[PubMed\]](#)
32. Wang, X.; Liu, K.; Wang, D.; Wu, L.; Fu, Y.; Xie, X. Multi-level recommendation reasoning over knowledge graphs with reinforcement learning. In Proceedings of the ACM Web Conference 2022, Virtual Event, 25–29 April 2022; pp. 2098–2108.
33. Ma, T.; Huang, L.; Lu, Q.; Hu, S. Kr-gcn: Knowledge-aware reasoning with graph convolution network for explainable recommendation. *ACM Trans. Inf. Syst.* **2023**, *41*, 1–27. [\[CrossRef\]](#)
34. Wang, X.; He, X.; Cao, Y.; Liu, M.; Chua, T.S. Kgat: Knowledge graph attention network for recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 950–958.
35. Yang, Y.; Huang, C.; Xia, L.; Li, C. Knowledge graph contrastive learning for recommendation. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022; pp. 1434–1443.
36. Wang, H.; Zhang, F.; Wang, J.; Zhao, M.; Li, W.; Xie, X.; Guo, M. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, 22–26 October 2018; pp. 417–426.
37. Peng, C.; Xia, F.; Naseriparsa, M.; Osborne, F. Knowledge graphs: Opportunities and challenges. *Artif. Intell. Rev.* **2023**, *56*, 13071–13102. [\[CrossRef\]](#)
38. Zhao, N.; Long, Z.; Wang, J.; Zhao, Z.D. AGRE: A knowledge graph recommendation algorithm based on multiple paths embeddings RNN encoder. *Knowl.-Based Syst.* **2023**, *259*, 110078. [\[CrossRef\]](#)
39. Wang, H.; Zhao, M.; Xie, X.; Li, W.; Guo, M. Knowledge graph convolutional networks for recommender systems. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 3307–3313.
40. Guo, Q.; Zhuang, F.; Qin, C.; Zhu, H.; Xie, X.; Xiong, H.; He, Q. A survey on knowledge graph-based recommender systems. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 3549–3568. [\[CrossRef\]](#)
41. Gambhir, M.; Gupta, V. Recent automatic text summarization techniques: A survey. *Artif. Intell. Rev.* **2017**, *47*, 1–66. [\[CrossRef\]](#)
42. El-Kassas, W.S.; Salama, C.R.; Rafea, A.A.; Mohamed, H.K. Automatic text summarization: A comprehensive survey. *Expert Syst. Appl.* **2021**, *165*, 113679. [\[CrossRef\]](#)
43. Jadhav, A.; Rajan, V. Extractive summarization with swap-net: Sentences and words from alternating pointer networks. In Proceedings of the ACL 2018—56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; pp. 142–151.
44. See, A.; Liu, P.J.; Manning, C.D. Get to the point: Summarization with pointer-generator networks. *arXiv* **2017**, arXiv:1704.04368.
45. Miller, D. Leveraging BERT for extractive text summarization on lectures. *arXiv* **2019**, arXiv:1906.04165.
46. Zhang, W.; Li, X.; Deng, Y.; Bing, L.; Lam, W. A survey on aspect-based sentiment analysis: Tasks, methods, and challenges. *IEEE Trans. Knowl. Data Eng.* **2022**, *35*, 11019–11038. [\[CrossRef\]](#)
47. Yang, H.; Zeng, B.; Yang, J.; Song, Y.; Xu, R. A multi-task learning model for chinese-oriented aspect polarity classification and aspect term extraction. *Neurocomputing* **2021**, *419*, 344–356. [\[CrossRef\]](#)
48. Yang, H.; Zhang, C.; Li, K. PyABSA: A Modularized Framework for Reproducible Aspect-based Sentiment Analysis. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, Birmingham, UK, 21–25 October 2023; pp. 5117–5122.
49. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.

50. Shrestha, A.; Mahmood, A. Review of deep learning algorithms and architectures. *IEEE Access* **2019**, *7*, 53040–53065. [[CrossRef](#)]
51. Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning entity and relation embeddings for knowledge graph completion. In Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; Volume 29.
52. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
53. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Philip, S.Y. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4–24. [[CrossRef](#)]
54. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
55. He, R.; McAuley, J. 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, Montreal, QC, Canada, 11–15 April 2016; pp. 507–517.
56. McAuley, J.; Targett, C.; Shi, Q.; Van Den Hengel, A. Image-based recommendations on styles and substitutes. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, 9–13 August 2015; pp. 43–52.
57. Ai, Q.; Azizi, V.; Chen, X.; Zhang, Y. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms* **2018**, *11*, 137. [[CrossRef](#)]
58. He, X.; Chua, T.S. Neural factorization machines for sparse predictive analytics. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Japan, 7–11 August 2017; pp. 355–364.
59. Rendle, S. Factorization machines. In Proceedings of the 2010 IEEE International Conference on Data Mining, Sydney, NSW, Australia, 13–17 December 2010; pp. 995–1000.
60. He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; Wang, M. 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, Virtual, 25–30 July 2020; pp. 639–648.
61. Tudi, M.R.; Na, J.C.; Liu, M.; Chen, H.; Dai, Y.; Yang, L. Aspect-Based Sentiment Analysis of Racial Issues in Singapore: Enhancing Model Performance Using ChatGPT. In Proceedings of the International Conference on Asian Digital Libraries, Taipei, Taiwan, 4–7 December 2023; pp. 41–55.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.