An Aspect Performance-aware Hypergraph Neural Network for Review-based Recommendation

Junrui Liu
liujunrui@emails.bjut.edu.cn
College of Computer Science, Beijing University of Technology
Beijing, China

Tong Li*
litong@bjut.edu.cn
College of Computer Science, Beijing University of Technology
Beijing, China

Di Wu
wudi7432@gmail.com
Beijing Police College
Beijing, China

Zifang Tang
zifangtang@emails.bjut.edu.cn
College of Computer Science, Beijing University of Technology
Beijing, China

Yuan Fang
yfang@smu.edu.sg
School of Computing and Information Systems, Singapore Management University
Singapore

Zhen Yang
yangzhen@bjut.edu.cn
College of Computer Science, Beijing University of Technology
Beijing, China

ABSTRACT

Online reviews allow consumers to provide detailed feedback on various aspects of items. Existing methods utilize these aspects to model users' fine-grained preferences for specific item features through graph neural networks. We argue that the performance of items on different aspects is important for making precise recommendations, which has not been taken into account by existing approaches, due to lack of data. In this paper, we propose an aspect performance-aware hypergraph neural network (APH) for the review-based recommendation, which learns the performance of items from the conflicting sentiment polarity of user reviews. Specifically, APH comprehensively models the relationships among users, items, aspects, and sentiment polarity by systematically constructing an aspect hypergraph based on user reviews. In addition, APH aggregates aspects representing users and items by employing an aspect performance-aware hypergraph aggregation method. It aggregates the sentiment polarities from multiple users by jointly considering user preferences and the semantics of their sentiments, determining the weights of sentiment polarities to infer the performance of items on various aspects. Such performances are then used as weights to aggregate neighboring aspects. Experiments on six real-world datasets demonstrate that APH improves MSE, Precision@5, and Recall@5 by an average of 2.30%, 4.89%, and 1.60% over the best baseline. The source code and data are available at https://github.com/dianziliu/APH.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

User review, Aspect, Hypergraph, Sentiment polarity aggregation

ACM Reference Format:
Junrui Liu, Tong Li, Di Wu, Zifang Tang, Yuan Fang, and Zhen Yang. 2025. An Aspect Performance-aware Hypergraph Neural Network for Review-based Recommendation. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining (WSDM '25), March 10–14, 2025, Hannover, Germany.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3701551.3703528

1 INTRODUCTION

Recommender systems have been extensively integrated into web services to enhance user experiences. Users are encouraged to share their feelings through ratings and reviews. Reviews contain users' opinions or sentiments about special aspects, where an aspect is a word or phrase describing a property of items and explicitly describing the characteristics of the items the user cares about [11, 20]. For example, in the sentence that *"Amazing sound and quality, all in one headset", "sound"* and *"quality"* are two aspects. As a type of important user-generated content, reviews can help recommender systems understand user preferences and item features [5, 21, 25, 35].

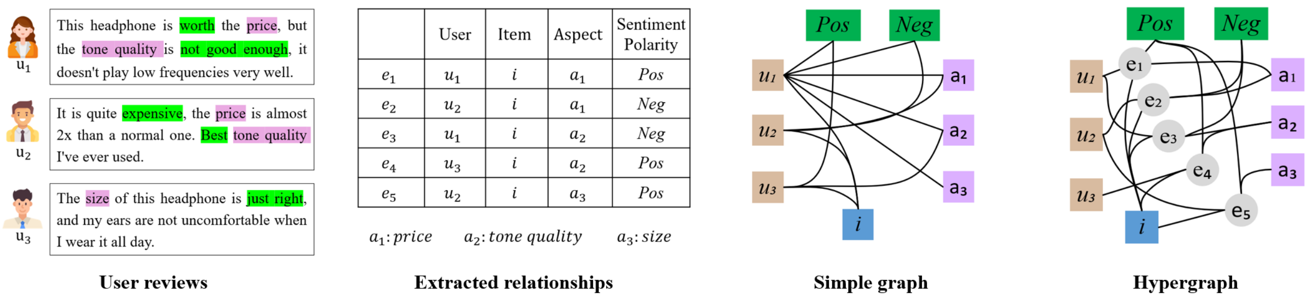
Existing methods detect aspects in user reviews and leverage them to model users' fine-grained preferences to specific item features by graph neural networks. For example, the APRE model [20] identifies the importance of aspects by considering the similarity between the aspects and their content features in reviews. MA-GNNs [41] constructs multiple aspect-aware user-item graphs and utilizes a routing-based fusion mechanism to allocate weights to different aspects. RGNN [26] regards aspects and sentiments as nodes and builds a subgraph for each user and item. It employs a type-aware graph attention mechanism that aggregates the context information from neighboring nodes to learn the node embeddings. A personalized graph pooling operator is proposed to learn the semantic representation for each user/item from the graph.

It is noteworthy that when users select items, they tend to prioritize item performances on various aspects. However, not all the performances can be directly obtained. This results in existing methods only considering user preferences in aspects reflected in the

[*] Corresponding author.

WSDM '25, March 10–14, 2025, Hannover, Germany

ACM ISBN 979-8-4007-1329-3/25/03
https://doi.org/10.1145/3701551.3703528



u_1: This headphone is worth the price, but the tone quality is not good enough, it doesn't play low frequencies very well.

u_2: It is quite expensive, the price is almost 2x than a normal one. Best tone quality I've ever used.

u_3: The size of this headphone is just right, and my ears are not uncomfortable when I wear it all day.

	User	Item	Aspect	Sentiment Polarity
e_1	u_1	i	a_1	*Pos*
e_2	u_2	i	a_1	*Neg*
e_3	u_1	i	a_2	*Neg*
e_4	u_3	i	a_2	*Pos*
e_5	u_2	i	a_3	*Pos*

a_1: *price* a_2: *tone quality* a_3: *size*

User reviews **Extracted relationships** **Simple graph** **Hypergraph**

Figure 1: Hypergraph vs. simple graph. There are three reviews written by three users for one headphone. Based on these reviews, we extract five relationships that record the users' sentiment polarity towards various aspects of the item. Formally, an vertex set is $\mathcal{V} = \{u_1, u_2, u_3, i, a_1, a_2, a_3, Pos, Neg\}$ and a relationship set is $\mathcal{E} = \{e_1, e_2, e_3, e_4, e_5\}$. In the simple graph, two vertices are joined together by an edge if they commonly exist in any relationship. This graph cannot tell us much information, like whether a user has a positive sentiment for what. In the hypergraph, each hyperedge e_n connects four vertices, and can completely illustrate one extracted relationship.

reviews when calculating the importance of aspects, without considering the actual performance of items in those aspects, leading to suboptimal results. We plan to extract and learn the performance of items on different aspects from user reviews. The reviews encompass users' opinions on a specific aspect of an item, and sentiment polarities in these opinions partially reflect the item's performance in that aspect. Nevertheless, there are conflicts in the sentiment polarities expressed by different users, which makes it challenging to accurately extract the item's performance in a given aspect from the reviews.

In reality, the conflicting sentiment polarities arise from the users' preferences in both aspects and sentiments. Therefore, to identify the true relationship between items and aspects, it is essential to consider user preferences when aggregating sentiment polarities. For example, users with light tastes think a dessert has just the right amount of sweetness, while for a dessert lover, it's too light. Combining the feedback from multiple users and their preferences, we can assume that this dessert is on the low side of sweetness.

In this paper, we propose an aspect performance-aware hypergraph neural network (APH) for the review-based recommendation, which learns the performance of items from the conflicting sentiment polarity of user reviews. Specifically, we first extract aspects and sentiment polarities from reviews to systematically construct an aspect-based hypergraph. Since the representation power of simple graphs is limited and cannot express the relationships between users, items, aspects, and sentiment polarities, we choose hypergraphs with stronger representation ability to model the relationships [8]. Figure 1 compares the representation abilities of these two types of graphs. Then, we design an aspect performance-aware hypergraph aggregation method that aggregates conflicting sentiment polarities to learn the performances of items on different aspects and treats them as weights in aggregating aspect nodes to represent the item. The sentiment polarity contained in the reviews is somewhat subjective due to user preferences, leading to conflicting sentiment polarities between different users. To accurately learn the performance of an item on an aspect, we aggregate multiple hyperedges related to the item and the aspect, and then assign weights to each aspect based on the performance using the attention mechanism. Furthermore, users may decide whether to buy an item because of its extreme performance on an aspect or the item's overall performance. Thus, to model the role of aspects in a user-item interaction, an aspect fusion layer aggregates the first-order aspect nodes of users and the items, respectively, to obtain the final vector representations used for prediction. Finally, a factorization machine (FM) is used to predict.

The main contributions of this paper are as follows:

- We propose an aspect performance-aware hypergraph neural network (APH) for the review-based recommendation, which both considers user preference for aspects and the performances of items in those aspects.
- We propose an aspect performance-aware hypergraph aggregation method that learns the performances of items on different aspects from conflicting emotional polarities.
- Experiments on six real-world datasets demonstrate that APH improves MSE, Precision@5, and Recall@5 by an average of 2.30%, 4.89%, and 1.60% over the best baseline.

2 RELATED WORK

In this section, we review rating prediction tasks in recommender systems and review-based recommendation methods. At the same time, we also discuss aspect extraction methods and graph neural networks, which are related to our methods.

2.1 Rating Prediction

Rating prediction is one of the most critical tasks for recommender systems and is widely known to researchers by Netflix Prize competition [17]. Slight performance enhancements of predictions could significantly improve the recommendations [2, 16]. Matrix factorization (MF) is a successful and recognized latent factor model that attracts much attention. The variations of MF includes PMF [28], SVD [17], FM [27], etc. With the great success of deep learning in many fields, researchers tend to apply deep learning techniques to enhance the performance of rating prediction tasks, such as NCF [12].

By the restrictions of rating sparsity, researchers employ additional information to enhance the prediction, such as user reviews [26, 35]. User reviews have a strong intrinsic correlation with user interests as they express their views on items through words. Analyzing the information in user reviews can provide insight into their logic and perception. Numerous studies have attempted to enhance the prediction performance of the rating prediction approaches by employing additional information from review texts [26, 35].

2.2 Review-based Recommendation

Many review-based models are proposed to improve the performance of rating prediction. These methods can be divided into two categories. The first category focuses on modeling the latent features in reviews. The second category models the interactions between users and items with review-based representation.

In this first category, some methods use natural language process (NLP) techniques to extract high-level features [35, 36]. ConvMF [15] and DeepCoNN [42] use CNN to extract local semantics. CARL [38], DAML [21] employs the local and mutual attention of CNN to learn features from reviews, and then integrates them with the latent factor model for rating prediction. TAERT [10] uses three attention networks to model different features, i.e., word contribution, review usefulness, and latent factors. RGNN [26] builds a review graph for each user where nodes are words and edges are word orders. Besides, some methods focus on fine-grained features, including explicit and implicit aspects [9, 20]. ANR [6] learns aspect-based representations for the user and item by an attention-based module. Moreover, the co-attention mechanism is applied to the user and item importance at the aspect level. CAPR [19] and ARPM [18] perform aspect and sentiment analysis on textual reviews and then establish users' and items' preference feature vectors. APRE [20] uses dependency parsing to extract explicit aspects and CNN to model user preferences based on them. In some literature, implicit aspects-based methods are regarded as high-level features-based methods [9]. MRCP [22] extracts word-level, review-level, and aspect-level features to represent users and items via a three-tier attention network. SENGR [31] is a sentiment-enhanced neural graph method that incorporates the information derived from textual reviews and bipartite graphs.

The second category models the interactions between users and items with review-based representation. D-attn [29] uses dual local and global attention to model word-level and review-level features. As global attention is applied to both the user side and the item side, it learns the interaction features between the two sides. Then the resultant factors are used for rating prediction, similar to matrix factorization. NARRE [3] filters useless reviews by using the vector representing each user and item as a part of attention scores. HTI [37] captures interactions based on reviews by mutually propagating textual features. Further, rather than representing users and items with static latent features, HTI dynamically identifies informative textual features at both word and review levels for each specific user-item pair. NRCA [23] points out two main paradigms of reviews, i.e., the document level and the review level. It uses a cross-attention mechanism to aggregate the informative words and reviews and represent users. DSRLN [25] extracts static and dynamic user interests by stacking attention layers that deal with sequence features and attention encoding layers that deal with of user-item interaction. Similar to Transformers [14], DSRLN adds temporal dependencies on sequence features.

2.3 Aspect Extraction and Sentiment Analysis

The Aspect extraction and sentiment Analysis task aims to extract aspect term, opinion term, and their associated sentiment. Existing methods are divided into two categories, supervised methods [4, 40], and unsupervised methods [7, 24]. Manually annotating data for training, which requires the hard labor of experts, is only feasible on small datasets in particular domains such as *Laptop* and *Restaurant*, which leads to supervised methods unsuited to our situation. Thus, we mainly focus on unsupervised methods. Hu and Liu [13] extracted the nouns/noun phrases from sentences, and such nouns/noun phrases were labeled as aspects. Once all aspects were selected, the nearest adjectives were extracted as potential opinion words. Some methods tend to identify the dependency of each word and design some rules to extract aspects [7, 24, 30]. Considering the impact of extracting aspect sentiment pairs on recommendation performance, we have chosen an unsupervised approach for extraction.

2.4 Graph Neural Networks in Recommendation

Graph neural networks (GNNs) extend deep learning techniques to process the graph data, and they are widely used in various fields [39]. Here we primarily focus on discussing GNN techniques used in reviews-based recommendation methods.

There are two main paradigms. One paradigm is the document-level that introduces features of reviews into a user-item graph. RGCL [33] constructs a review-aware user-item graph, where each edge feature is composed of both the user-item rating and the corresponding review semantic features. The feature-enhanced edges can help learn each neighbor node weight attentively for user and item representation learning. The other paradigm is word-level, i.e., regard words in reviews as graph nodes and then aggregates them to represent users/items. These methods mainly use graph attention networks to represent nodes by aggregating their neighbor nodes[1, 34]. The attention mechanism determines the weights of neighbor nodes. Li et al. [20] identify the importance of aspects by attention mechanism that regards the content features of aspects in reviews as the query to calculate the weights. The DualGCN model [32] regards aspects and sentiments as nodes in aspect graphs and adopts sum pooling to represent users and items. RGNN [26] builds a review graph for each user where nodes are words, and edges are word orders. It uses a type-aware method, which regards the combination of the type of edges and neighbor nodes as keys in the attention mechanism, to aggregate the information of neighboring words effectively. The MA-GNNs model [41] predefines four aspects and constructs multiple aspect-aware user-item graphs, regarding the aspect-based sentiment as the edge. It utilizes a routing-based fusion mechanism to allocate weights to different aspects, realizing the dynamic fusion of aspect preferences.

3 PRELIMINARIES

The hypergraph is a generalization of the graph [8]. Different from the graph, an edge in the hypergraph, called hyperedge, is a subset of all vertices in the hypergraph. A hypergraph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \phi)$, which includes vertex set $\mathcal{V}$, a hyperedge set $\mathcal{E}$, and a node type mapping function $\phi : \mathcal{V} \rightarrow \mathcal{T}$. Here $\mathcal{T}$ denotes the sets of predefined node types. A hypergraph $\mathcal{G}$ can be described by an $|\mathcal{V}| \cdot |\mathcal{E}|$ incidence matrix $\mathbf{H}$, whose entries are defined as $\mathbf{H}(v, e) = \begin{cases} 1 & if\ v \in e \\ 0 & otherwise \end{cases}$.

4 METHOD

To model user preferences for aspects and the performance of items in these aspects, APH extracts aspect-sentiment pairs from reviews. The overview of APH is shown in Figure 2. Specifically, APH has four steps for recommendation predictions: aspect hypergraph construction, aspect performance-aware hypergraph aggregation, aspect fusion, and prediction. The aspect hypergraph construction extracts aspect-sentiment pairs from reviews and then constructs the graph. Considering the user preference for aspects and the performance of items in those aspects, an aspect performance-aware hypergraph aggregation layer is designed to learn the aspect weights based on user sentiments. To learn more about the role of aspects in user-item interactions, we design the aspect fusion layer that aggregates neighboring aspects of users/items to represent users and items. The prediction layer fuses users and items to make predictions.

4.1 Aspect Hypergraph Construction

We aim to extract the explicit aspect, a word or phrase that describes the property of items [11, 20]. Through explicit aspects, recommender systems learn the fine-grained preferences that describe the special properties of items that users are interested in. Most existing supervised methods are trained based on *Laptop* and *Restaurant* datasets, which cannot fully meet our scenarios. Thus, we utilize a rule-based unsupervised method to extract aspects and sentiments from reviews [20].

The rule-based unsupervised method considers three dependency relations[1] sequentially used to extract candidate aspect-sentiment pairs, including *amod*, *nsubj+acomp*, and *dobj*. Table 1 summarizes how to extract aspects and sentiments based on dependency relations. On the one hand, some nouns directly describe the properties of items. They are considered potential aspects and are modified by adjectives with two types of dependency relations, *amod* and *nsubj+acomp*. The pairs of nouns and the modifying adjectives compose the AS-pair candidates. For instance, in the sentence *"Amazing sound and quality, all in one headset"*, the nouns *"sound"* and *"quality"* are two aspects of an item, and the user thinks the item in these aspects are amazing, which is a positive sentiment. On the other hand, the predicate and the object in a sentence describe the function of items. Thus, the combinations of predicates and objects are considered potential aspects by *dobj* dependency relation. In a combination, the predicate is regarded as the sentiment as it is usually with users' emotions and the whole combination is regarded as the aspect. For example, in the sentence *"If you're recording vocals this will eliminate the pops"*, the verb *"eliminate"* and noun *"pops"* construct a *dobj* relation. From this sentence, we know that the mentioned device is used to filter pops, and the word *"eliminate"* includes the user's emotions. Finally, to simplify the complexity of modeling, we use a sentiment analysis tool to judge the sentiment polarity of sentiment words [7]. Three kinds of positive emotions, neutral emotions, and negative emotions were extracted from sentiment words [2]. For more details please refer to the Appendix.

Table 1: Extracting aspects based on dependency relations

No.	Dependency relations	Aspect	Sentiment
1	Adj. (x) ← amod – Noun (y)	y	x
2	Noun (x) ←nsubj– Linking Verb (y) – – acomp → Adj. (z)	x	z
3	Verb (x) – dobj → Noun (y)	(x,y)	x

By the aspect-sentiment extraction method, we extract aspect-sentiment pairs for each review. After adding the context of users and items, we obtain quadruples. The form of a quadruple is (u, i, a, s), where u is a user, i is an item, a is an aspect, s is the sentiment polarity. Since hypergraphs have stronger expressive power than simple graphs, we construct hypergraphs based on quadruple [43]. Figure 1 compares the expressive power of hypergraphs and simple graphs. In our graph, the set of predefined node types, $\mathcal{T}$, contains four types, i.e., user $\mathcal{U}$, item $\mathcal{I}$, aspect $\mathcal{A}$, and sentiment polarity $\mathcal{S}$. Each hyperedge contains a user, an item, an aspect, and a sentiment polarity that is the same as the quadruple.

4.2 Aspect Performance-aware Hypergraph Aggregation

Formally, $\mathcal{A}_i$ is the aspect set associated with item i. Existing methods [20, 32, 41] aggregate aspects to represent items, is as follows:

$$\begin{aligned} \mathbf{x}_i &= f(\mathcal{A}_i) \\ &= \sum_{a \in \mathcal{A}_i} w(\mathbf{x}_a) \cdot \mathbf{x}_a, \end{aligned} \tag{1}$$

where $w(x)$ is a weight function and is usually a softmax function, $\mathbf{x}_a$ is the embedding vector of the aspect a. However, Equation (1) overlooks the user's demands for aspects and the performance of items in these aspects. It should be re-write as

$$\begin{aligned} \mathbf{x}_i &= f(\mathcal{A}_i) \\ &= \sum_{a \in \mathcal{A}_i} w(\mathbf{x}_a, p_i(a)) \cdot \mathbf{x}_a, \end{aligned} \tag{2}$$

where $p_i(a)$ is a performance metric function to denote the performance of item i on the aspect a.

Learning the performance metric function q is challenging in practical situations. We lack data about the performance of items on various aspects but have a lot of conflicting user sentiments about them. Users' sentiment not only indicates that they care about certain aspects but also reflects whether the performance of the items aligns with their needs in this aspect. However, sentiment polarities expressed by different users are conflicting. The subjective feelings encompassed in user sentiments may not accurately describe the relationships between different aspects of items. Therefore, the

[1] We use spaCy(https://spacy.io) to extract the syntactic relations between the words.

[2] We use the Opinion Lexicon, which is available at https://www.cs.uic.edu/~lzhang3/programs/OpinionLexicon.html.

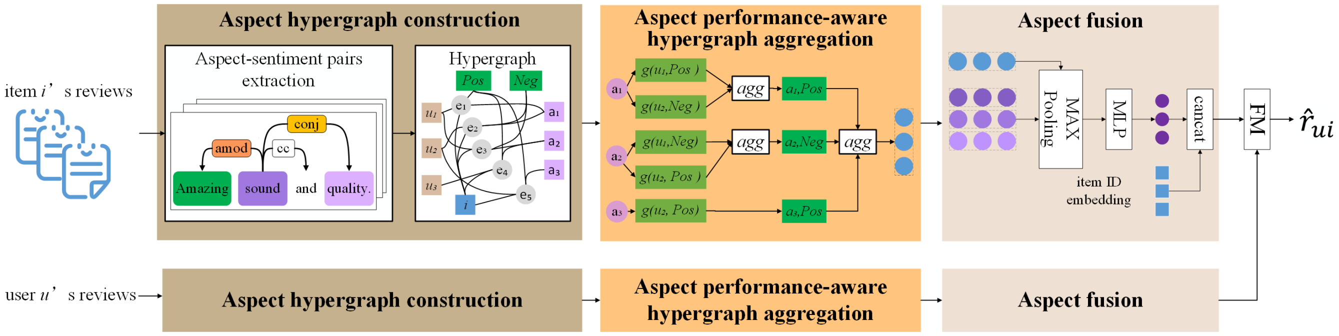


Figure 2: Framework of APH. It first extracts aspects and user sentiments from reviews to construct a hypergraph. Then, to learn the true relationship between an item and an aspect from conflicting user sentiments, APH considers user preferences to identify the weight of their sentiments. Likewise, we use a similar way to calculate the aspect-based user representations. Finally, APH fuses items, neighbor aspect nodes, and their ID embeddings to make predictions.

method should consider user preferences in the aggregation process to identify the true relationships between items and aspects. $\mathcal{U}_i$ is the user set that rated the item i, and $\mathcal{S}_i$ is the sequence list of users' sentiments on aspects. Our aggregation representation of an item is:

$$\begin{aligned}\mathbf{x}_i &= f(\mathcal{A}_i, q_i(\mathcal{S}_i, \mathcal{U}_i))\\ &= \sum_{a\in\mathcal{A}_i}\ \sum_{u,s\in\mathcal{U}_i,\mathcal{S}_i} w(\mathbf{x}_a, q_i(\mathbf{x}_u, \mathbf{x}_s))\cdot \mathbf{x}_a,\end{aligned} \tag{3}$$

where the function q is used to acquire the aspect performance metric function from user preferences and sentiments. It aggregates the sentiment polarities of multiple users by jointly considering their preferences and the semantic meaning, thereby determining the weights of the sentiment polarities.

We extend the aforementioned approach to the hypergraph we have constructed. $\mathcal{E}_i$ is the set of hyperedges that connected with an item i, and $\mathcal{E}_{i,a}$ it the subset with an aspect a. For each hyperedge $e \in \mathcal{E}_{i,a}$, its weight is:

$$\begin{aligned}w(e) &= w(u,i,a,s)\\ &= \frac{exp[\pi(\mathbf{x}_i, q_i(\mathbf{x}_u, \mathbf{x}_s), \mathbf{x}_a)]}{\sum_{e'\in\mathcal{E}_i} exp[\pi(\mathbf{x}_i, q_i(\mathbf{x}_{u'}, \mathbf{x}_{s'}), \mathbf{x}_a)]},\end{aligned} \tag{4}$$

where $\mathbf{x}_u,\ \mathbf{x}_i,\ \mathbf{x}_a,\ \mathbf{x}_s \in \mathbb{R}^{1\times d_1}$ are the input embeddings of nodes. Note that the calculation range of the softmax function is $\mathcal{E}_i$ instead of $\mathcal{E}_{i,a}$. The former can obtain the weight of each edge according to the performance difference of different aspects, while the latter makes the weight of each aspect equal to 1 after aggregating the edges related to the aspect.

Considering the nonlinear relationship between sentiment features and user preferences, we use MLP as the implementation of the function q_i. The result is denoted as $\mathbf{x}_q \in \mathbb{R}^{1\times d_1}$ and is following:

$$\mathbf{x}_q = q_i(\mathbf{x}_u, \mathbf{x}_s) = MLP(\mathbf{x}_u, \mathbf{x}_s), \tag{5}$$

$\pi(\mathbf{x}_i, \mathbf{x}_q, \mathbf{x}_a)$ is implemented by the following relational attention mechanism:

$$\pi(\mathbf{x}_i, \mathbf{x}_q, \mathbf{x}_a) = LeakyRelu[(\mathbf{x}_i\mathbf{W}_1)(\mathbf{x}_q\mathbf{W}_2 + \mathbf{x}_a\mathbf{W}_3)] \tag{6}$$

$\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3 \in \mathbb{R}^{d_1\times d_2}$ are three different weight matrices used for the linear transformations of the node edge embedding, and d_2 denotes the dimension of the hidden space of the graph representation learning layers. Finally, we aggregate all aspects in $\mathcal{E}_i$ to represent the item i, as following

$$\hat{\mathbf{x}}_i = \sum_{\mathcal{E}_{i,a}\in\mathcal{E}_i}\ \sum_{e\in\mathcal{E}_{i,a}} w(e)\mathbf{x}_a\mathbf{W}_4, \tag{7}$$

where $\mathbf{W}_4 \in \mathbb{R}^{d_1\times d_2}$ denotes the transform matrix. Likewise, we use a similar way to calculate the aggregation representation $\hat{\mathbf{x}}_u$ of a user u.

4.3 Aspect Fusion

Users may decide whether to buy an item because of its extreme performance in an aspect or the item's overall performance. To learn more about the role of aspects in user-item interactions, we have designed the aspect fusion layer that aggregates neighboring aspects of users/items to represent users and items. For an item i, their aspect neighbors $\mathcal{A}_i$'s feature matrix is denoted as $\mathbf{X}_i \in \mathbb{R}^{|\mathcal{A}_i|\times d_2}$. The rows in $\mathbf{X}_i$ is ranked under Equation (6). We use the max-pooling and MLP to generate a representation as follows,

$$\hat{\mathbf{g}}_i = max_{t=1}^{d_2}\mathbf{X}_i(:,t), \tag{8}$$

$$\mathbf{g}_i = ReLU(\hat{\mathbf{g}}_i\mathbf{W}_6 + b_6), \tag{9}$$

where t is a hyperparameter that determines the number of aspects that are aggregated, $\mathbf{W}_6 \in \mathcal{R}^{d_2\times d_2}$ and $b_6 \in \mathcal{R}^{1\times d_2}$ the weight matrix and bias vector. Then, we concatenate the non-linear transform of the item embedding $\mathbf{x}_i \in \mathcal{R}^{1\times d_1}$ after a MLP layer and the representation of sentiment-aware aggregation to generate the aspect-aware representation of the item i $\mathbf{q}_i$ as follows,

$$\mathbf{m}_i = ReLU(\hat{\mathbf{x}}_i\ \mathbf{W}_7 + b_7), \tag{10}$$

$$\mathbf{y}_i = \mathbf{m}_i \oplus \mathbf{g}_i, \tag{11}$$

where $\oplus$ is the concatenating operation, $\mathbf{W}_7 \in \mathcal{R}^{d_1\times d_2}$ and $b_7 \in \mathcal{R}^{1\times d_2}$ are the weight matrix and bias vector. Similarly, we can get the semantic representation of a user u as $\mathbf{y}_u$.

4.4 Prediction

An FM layer is used to predict final scores [27]. It considers the higher-order interactions between the user and item fine-grained features. Specifically, we first concatenate the user and item representations as $\mathbf{z} = \mathbf{y}_u \oplus \mathbf{y}_i$, and the prediction r_{ui} is defined as follows

$$\hat{r}_{ui} = b_0 + b_u + b_i + \mathbf{z}\,\mathbf{w}^T + \sum_{i=1}^{d'} \sum_{j=i+1}^{d'} < \mathbf{v}_i, \mathbf{v}_j > \mathbf{z}_i\,\mathbf{z}_j, \qquad (12)$$

where b_0, b_u, and b_i are the global bias, user bias, and item bias, respectively. $\mathbf{w} \in \mathcal{R}^{1\times d'}$ is the coefficient vector, and $d' = 4 \times d_2$. $\mathbf{v}_i, \mathbf{v}_j \in \mathcal{R}^{1\times d'}$ are the latent factors associated with i-th and j-th dimension of $\mathbf{z}$. z_i is the value of the i-th dimension of $\mathbf{z}$. The model parameters Φ of APH can be learned by solving the following optimization problem,

$$\mathcal{L} = \min \frac{1}{|D|} \sum_{u,i\in D} (r_{ui} - \hat{r}_{ui})^2 + \lambda ||\Phi||_F^2, \qquad (13)$$

where λ is the regularization parameter, and D denotes the set of user-item pairs used to update the model parameters.

4.5 Time complexity

APH mainly has three parts, hypergraph aggregation (HA), aspect fusion (AF), and FM for prediction. For each prediction, the time complexity of HA is $O_{HA}(|\mathcal{E}_i|d^3)$, where $|\mathcal{E}_i|$ is the number of hyperedges item i connected, that of AF is $O_{AF}(d^2)$, and that of FM is $O_{AF}(d^2)$. Summary, the time complexity of APH is $O(|\mathcal{E}_i|d^3)$ for each user-item pair.

5 EXPERIMENT

In this section, we do a series of experiments to identify the performance of our method. We describe our experimental setup and show comparison results with different baselines. We further use an ablation study to identify the effect of each part in APH. Finally, we analyze the extracted aspects. For more experiments, such as hyperparameter analysis, please refer to the Appendix.

5.1 Dataset

The experiments are performed on the Amazon review and Yelp datasets, which have been widely used for recommendation research [21, 26]. For the Amazon review dataset, we choose the following 5-core review subsets for evaluation: Musical Instruments, Office Products, Toys and Games, Video Games, and Beauty (respectively denoted by Music, Office, Toys, Games, and Beauty). For the Yelp dataset, we only keep the users and items that have at least 10 reviews for experiments. Table 2 summarizes the details of these experimental datasets. We discuss the extract aspects in section 5.6.

Table 2: The statistics of the experimental datasets.

Dataset	#Users	#Items	#Ratings/Reviews	#Density
Music	1,429	900	10,261	0.80%
Office	4,905	2,420	53,228	0.45%
Toys	19,412	11,924	167,597	0.07%
Games	24,303	10,672	231,577	0.09%
Beauty	22,363	12,101	198,502	0.07%
Yelp	26,084	65,786	3,519,533	0.04%

Table 3: The performances of different recommendation methods evaluated by MSE. The best results are in bold faces and the second-best results are underlined. * indicates that the standard deviation of the results of the five times is smaller than 0.001.

Dataset	Music	Office	Toys	Games	Beauty	Yelp
PMF	1.8783	0.9635	1.6091	1.5260	2.7077	1.4217
SVD++	0.7952	0.7213	0.8276	1.2081	1.2129	1.2973
CDL	1.2987	0.8763	1.2479	1.6002	1.7726	1.4042
DCN	0.7909	0.7315	0.8073	1.1234	1.2210	1.2719
NARRE	0.7688	0.7266	0.7912	1.1120	1.1997	1.2675
CARL	0.7632	0.7193	0.8248	1.1308	1.2250	1.3199
DAML	0.7401	0.7164	0.7909	1.1086	1.2175	1.2700
NRCA	0.7658	0.7343	0.8100	1.1259	1.2034	1.2721
DSRLN	0.7538	0.7131	0.8141	1.1205	1.1951	<u>1.1655</u>
ANR	0.7825	0.7237	0.7974	1.1038	1.2021	1.2708
RGNN	<u>0.7319</u>	<u>0.7125</u>	**0.7786**	<u>1.0996</u>	<u>1.1885</u>	1.2645
APH	**0.6795***	0.**6884***	<u>0.7859</u>*	**1.0829**	**1.1757***	**1.1467***

5.2 Baselines

We compare APH with three types of baselines. Traditional rating-based methods include:PMF [28], SVD++ [17]. Review-based methods include: CDL [36], DeepCoNN [42], NARRE [3], ANR [6], CARL [38], DAML [21], and DSRLN [25]. Aspect-based methods include: NRCA [23], MA-GNNs [41], and RGNN [26]. These methods have been discussed in Section 2. As MA-GNNs is trained by pairwise loss, we only compared it with NDCG.

5.3 Setup

For each dataset, we randomly choose 20% of the user-item review pairs (denoted by D_{test}) for evaluating the model performance in the testing phase, and the remaining 80% of the review pairs (denoted by D_{train}) are used in the training phase.

In recommender systems, there are two common tasks: rating prediction and click-through rate prediction. Thus, to evaluate the performance of our method in the rating prediction task, we apply the typically used Mean Square Error (MSE) and Normalized Discounted Cumulative Gain (NDCG), which has been widely used in previous studies [26, 38, 42]; we use Precision(Pre) and Recall(Rec) for click-through rate prediction to evaluate the Top-K performance.

5.4 Performance Comparison

5.4.1 Model performance on rating prediction task. The MSE and NDCG results of the performance comparison are shown in Table 3 and Table 4, respectively. We mark the best results in bold faces and the second-best results are underlined. APH achieves the best results compared with other baselines in five datasets and the second-best results in the remaining one. On average, APH improves MSE by 2.30% compared to the best baseline. Nevertheless, our approach has achieved an improvement in NDCG. These results show that APH can effectively improve prediction performance by modeling the performance of items in aspects. Explicit aspects describe the fine-grained preferences of users and explain

Table 4: The performances of different recommendation methods evaluated by NDCG@10. The best results are in bold faces and the second-best results are underlined. * indicates that the standard deviation of the results of the five times is smaller than 0.001.

Dataset	Music	Office	Toys	Games	Beauty	Yelp
DCN	0.977	0.973	0.975	0.971	0.966	0.941
NARRE	0.978	0.976	0.981	0.968	0.971	0.957
CARL	0.980	0.978	0.978	0.969	0.966	0.943
DAML	0.982	0.978	0.979	**0.979**	0.967	0.958
DSRLN	0.781	0.974	0.977	**0.979**	0.967	0.948
MA-GNNs	0.979	0.973	0.975	0.966	0.965	0.933
RGNN	0.982	0.983	0.982	0.976	0.973	0.963
APH	**0.988***	**0.986***	**0.983***	0.977*	**0.974***	**0.965***

the characteristics of the items that the user cares about. APH models fine-grained preferences from explicit aspect-sentiment pairs to enhance prediction performance. To drop out subjective feelings in user sentiments and identify the true relationships between items and aspects, APH designs an aspect performance-aware aggregation layer that separates the user preferences from the sentiment. Thus, APH effectively improves recommendation performance. In the Toys dataset, RGNN performs a better MSE result than APH. We observe that the most frequent aspects in the Toy dataset reflect the characteristics of the user, followed by the characteristics of the item, which is different from other datasets. This situation could amplify the variance of the prediction error. In summary, APH models user preferences for aspects and the performance of items in these aspects, enhancing recommendation performance.

5.4.2 Model performance on click-through rate prediction. For CTR prediction, we use cross-entropy loss to train all models and add a sigmoid layer as the activation function. The ratio of negative sampling is 4, i.e., we sample 4 negative items from unobserved items for each positive item. Other settings are the same as those for rating prediction. The numerical results on all the benchmark datasets are displayed in Table 5. APH achieves the best results compared with other baselines in six datasets. Compared to the base baseline, APH achieves an average improvement 4.89% on Pre@5 and 1.60% on Rec@5. For the CTR task, APH can more effectively distinguish the difference between positive and negative items than baseline, by learning the item's performance in certain aspects. The design of APH helps recommender systems recommend more accurately.

5.5 Ablation Study

To investigate the importance of each component of APH, we consider the following variants of APH for experiments:

- APH(MAX/MEAN) dropouts the aspect performance-aware aggregation layer and uses max/mean pooling instead.
- APH(-AF) dropouts the aspect fusion layer.
- APH(-FM) uses the dot function to predict ratings.

The experiment results are shown in Table 6. It determines that the aspect performance-aware aggregation layer and the aspect fusion layer positively impact performance. User sentiments contain users' subjective feelings and do not effectively describe the relationships between different aspects of items. Therefore, APH proposes an aspect performance-aware aggregation layer, which learns the performance of items in aspects from user sentiments. We use the max and mean pooling layers instead of the aggregation layer, leading the model to fail to determine the performances of items on the aspects. Despite introducing the aspect performance-aware aggregation layer effectively identifying the importance of aspects in the hypergraph, achieving satisfactory results during the prediction phase still relies on successfully matching user demands with item performance. To learn more about the role of aspects in user-item interactions, we have designed the aspect fusion layer that aggregates neighboring aspects of users/items to represent users and items. From the results, we can find that the aspect fusion layer successfully aggregates aspects to represent users and items and matches them for predictions. In a word, these two layers are important for our method.

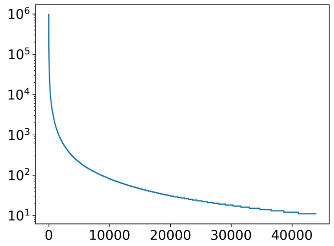


Figure 3: Aspect distribution in the Yelp dataset, which is similar to the other five datasets.

5.6 Analysis of Extracted Aspect

Our method aggregates explicit aspects to represent users and items and fuses them to make predictions. The aspects are extracted by an unsupervised method, discussed in section A. In this subsection, we show the statistics of explicit aspects in Table 7. The number of aspects is smaller than that of items, and the number of quadruples is bigger than that of ratings. These characteristics can help to reduce the size of the parameter space. We also give the distribution of aspects in Figure 7. Most distributions are long-tail distributions. This paper focuses on the impact of aspects, so the impact of distributions will be studied in the future. Top-10 explicit aspects in various datasets give an overview of the quality of extracted aspects, which are shown in Table 8. We can see that the extracted aspects include some normal terms, like "quality", "color", "price", and some special terms like "amp", "skin", etc. It determines that our method can effectively extract explicit aspects from reviews. In some situations, our method faces trouble and regards "son" and "daughter" as aspects that reduce model performance, which leads our method to play the second performance in this dataset.

5.7 Case study

In addition, we also conduct a case study to explore whether APH learns the performance of items on an aspect. We randomly select an item "*B*0000538*AC*" from the Office dataset that has conflicting sentiment polarities from different users. Figure 4 visualizes related aspect quadruples, the subgraph of the item, the attention scores of user sentiment polarities calculated by Equation 4, and

Table 5: The performances of different recommendation methods evaluated by P@5 and R@5. The best results are in bold faces and the second-best results are underlined. * and ‡ indicate that the Standard Deviation of the results of the five times is smaller than 0.001 and 0.002, respectively.

	Music		Office		Toys		Games		Beauty		Yelp	
	Pre@5	Rec@5	Pre@5	Rec@5	Pre@5	Rec@5	Pre@5	Rec@5	Pre@5	Rec@5	Pre@5	Rec@5
DCN	0.2327	0.6818	0.2555	0.5953	0.2408	0.6228	0.2561	0.6355	0.2876	0.7024	0.3238	0.5985
NARRE	0.2502	0.6603	0.3265	0.7361	0.0105	0.0341	0.2053	0.4984	0.1545	0.4094	0.3976	0.5907
DAML	0.2515	0.7019	0.3158	0.6796	0.2517	0.6638	0.2598	0.6622	0.2227	0.5911	0.3861	0.6138
RGNN	0.2690	0.7453	0.3229	0.6967	0.2874	0.7599	0.2809	0.7164	0.2985	0.7387	0.3824	0.6592
DSRLN	0.2721	0.7518	0.3386	0.7386	0.2873	0.7503	0.2673	0.7131	0.3044	0.7642	0.4278	**0.7248**
APH	**0.2730***	**0.7566**‡	**0.3461***	**0.7433**‡	**0.2985** *	**0.7614***	**0.3263***	**0.7890***	**0.3158***	**0.7753***	**0.4407***	0.6996*

Table 6: MSE results of ablation study.

Dataset	Music	Office	Toys	Games	Beauty	Yelp
APH(MAX)	0.7006	0.6951	0.7972	1.0879	1.1773	1.1913
APH(MEAN)	0.6933	0.7010	0.7918	1.0799	1.1820	1.1755
APH(-AF)	0.6873	0.7068	0.8040	1.0958	1.1899	1.1869
APH(-FM)	0.8173	0.7196	0.8228	1.1052	1.1999	1.1714
APH	**0.6795**	**0.6884**	**0.7859**	**1.0829**	**1.1757**	**1.1467**

Table 7: The statistics of explicit aspects in various datasets.

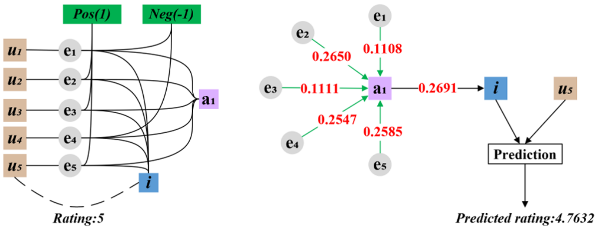
Dataset	# Aspect	# Quadruple
Music	601	38,898
Office	3,092	393,038
Toys	4,809	776,819
Games	11,656	2,439,534
Beauty	4,868	866,835
Yelp	43,904	20,857,681

Table 8: Top-10 explicit aspects in various datasets.

Music	Office	Toys	Games	Beauty	Yelp
quality	quality	toy	back	hair	place
guitar	mark	kid	graphic	product	food
draw	color	part	way	scent	service
good	printer	daughter	thing	skin	staff
price	product	boy	quality	color	restaurant
one	price	quality	level	have_hair	selection
wheel	part	back	point	price	price
thing	paper	one	part	have_skin	portion
base	thing	fit	work	face	experience
amp	size	thing	control	smell	sauce

	User	Item	Aspect	Sentiment polarity
e_1	*A2582KMXLK2P06*	*B0000538AC*	*pack*	*Neg*
e_2	*A156P4FPL8OGXB*	*B0000538AC*	*pack*	*Pos*
e_3	*A3S15YGZ6W6EV2*	*B0000538AC*	*pack*	*Pos*
e_4	*A1S7BFT0HDF3HA*	*B0000538AC*	*pack*	*Neg*
e_5	*A3QS4WWC1LCA6H*	*B0000538AC*	*pack*	*Pos*

(a) Extracted aspect quadruples.



(b) The subgraph of the item.

(c) Intermediate results of the model.

Figure 4: A case study. We show the extracted aspect quadruples of an item "*B0000538AC*" and an aspect "pack"; then we build the subgraph; we also show the attention scores calculated by Equation (4), and the final predicted rating.

the final predictive rating. APH considers the item's performance of the "pact" aspect to be 0.2691, higher than the mean of the sentiment polarities($Neg = -1, Pos = 1$). When the aggregation aspects represent an item, the aspect performance-aware hypergraph aggregation layer calculates the performance of the item in aspects based on the user's sentiment polarities, making the aggregation results more accurate.

6 CONCLUSION

Due to the performances of items on aspects being unavailable in datasets, existing methods only consider user preferences in aspects reflected in the reviews when aggregating aspects, and do not consider the actual performance of items in those aspects, leading to suboptimal results. We argue that the performances can be extracted and learned from user reviews. To this end, this paper proposes an aspect performance-aware hypergraph neural network for recommender systems, which considers user preference for aspects and the performance of items in those aspects when calculating their importance. We extract aspect-sentiment pairs from reviews and then construct an aspect-based hypergraph. Subsequently, we design a method that incorporates user preferences in aspect sentiment pairs to aid in aggregating conflicting sentiment features and learn the item's performance in each aspect. An aspect fusion layer respectively combines aspects with users and items, modeling the role that aspects play in the interaction between users and items. Experiments on six real-world datasets demonstrate that the predictions of APH significantly outperform baselines. In future work, we plan to extract aspect categories to enhance the connectivity of aspect graphs.

ACKNOWLEDGMENTS

This work is partially supported by the National Key R&D Program of China (No.2022YFB3103100), and the Major Research Plan of the National Natural Science Foundation of China (No.92167102).

REFERENCES

[1] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2014. Spectral Networks and Locally Connected Networks on Graphs. In *Proceedings of the 2nd ICLR Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings.*

[2] James Chambua and Zhendong Niu. 2021. Review text based rating prediction approaches: preference knowledge learning, representation and utilization. *Artif. Intell. Rev.* 54, 2 (2021), 1171–1200.

[3] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural Attentional Rating Regression with Review-level Explanations. In *Proceedings of the 27th WWW, Lyon, France, April 23-27, 2018.* 1583–1592.

[4] Hao Chen, Zepeng Zhai, Fangxiang Feng, Ruifan Li, and Xiaojie Wang. 2022. Enhanced Multi-Channel Graph Convolutional Network for Aspect Sentiment Triplet Extraction. In *Proceedings of the 60th ACL, Dublin, Ireland, May 22-27, 2022.* 2974–2985.

[5] Lu Cheng, Ruocheng Guo, and Huan Liu. 2022. Estimating Causal Effects of Multi-Aspect Online Reviews with Multi-Modal Proxies. In *Proceedings of the 15th WSDM, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022.* 103–112.

[6] Jin Yao Chin, Kaiqi Zhao, Shafiq R. Joty, and Gao Cong. 2018. ANR: Aspect-based Neural Recommender. In *Proceedings of the 27th CIKM, Torino, Italy, October 22-26, 2018.* 147–156.

[7] Mauro Dragoni, Marco Federici, and Andi Rexha. 2019. An unsupervised aspect extraction strategy for monitoring real-time reviews stream. *Inf. Process. Manag.* 56, 3 (2019), 1103–1118.

[8] Yue Gao, Zizhao Zhang, Haojie Lin, Xibin Zhao, Shaoyi Du, and Changqing Zou. 2022. Hypergraph Learning: Methods and Practices. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 5 (2022), 2548–2566. https://doi.org/10.1109/TPAMI.2020.3039374

[9] 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 Trans. Inf. Syst.* 37, 3 (2019), 28:1–28:27.

[10] Siyuan Guo, Ying Wang, Hao Yuan, Zeyu Huang, Jianwei Chen, and Xin Wang. 2021. TAERT: Triple-Attentional Explainable Recommendation with Temporal Convolutional Network. *Inf. Sci.* 567 (2021), 185–200.

[11] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An Unsupervised Neural Attention Model for Aspect Extraction. In *Proceedings of the 55th ACL, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers.* Association for Computational Linguistics, 388–397. https://doi.org/10.18653/V1/P17-1036

[12] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th WWW,Perth, Australia, 2017.* 173–182.

[13] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the 10th SIGKDD, Seattle, Washington, USA, August 22-25, 2004.* 168–177.

[14] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *Proceedings of the 18th ICDM, Singapore, November 17-20, 2018.* 197–206.

[15] Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. 2016. Convolutional matrix factorization for document context-aware recommendation. In *Proceedings of the 10th ACM conference on recommender systems.* 233–240.

[16] Yehuda Koren. 2018. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the SIGKDD 2008, Las Vegas, Nevada, USA, August 24-27, 2008.* 426–434.

[17] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.

[18] Chin-Hui Lai and Chia-Yu Hsu. 2021. Rating prediction based on combination of review mining and user preference analysis. *Information Systems* 99 (2021), 101742.

[19] 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 SIGIR, Paris, France, July 21-25, 2019.* 275–284.

[20] Zeyu Li, Wei Cheng, Reema Kshetramade, John Houser, Haifeng Chen, and Wei Wang. 2021. Recommend for a Reason: Unlocking the Power of Unsupervised Aspect-Sentiment Co-Extraction. In *Proceedings of theEMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021.* Association for Computational Linguistics, 763–778.

[21] Donghua Liu, Jing Li, Bo Du, Jun Chang, and Rong Gao. 2019. Daml: Dual attention mutual learning between ratings and reviews for item recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining.* 344–352.

[22] Hongtao Liu, Wenjun Wang, Qiyao Peng, Nannan Wu, Fangzhao Wu, and Pengfei Jiao. 2021. Toward Comprehensive User and Item Representations via Three-tier Attention Network. *ACM Trans. Inf. Syst.* 39, 3 (2021), 25:1–25:22.

[23] Hongtao Liu, Wenjun Wang, Hongyan Xu, Qiyao Peng, and Pengfei Jiao. 2020. Neural Unified Review Recommendation with Cross Attention. In *Proceedings of the 43rd SIGIR, Virtual Event, China, July 25-30, 2020.* 1789–1792.

[24] Qian Liu, Bing Liu, Yuanlin Zhang, Doo Soon Kim, and Zhiqiang Gao. 2016. Improving Opinion Aspect Extraction Using Semantic Similarity and Aspect Associations. In *Proceedings of the 30th AAAI, February 12-17, 2016, Phoenix, Arizona, USA.* AAAI Press, 2986–2992.

[25] Tongcun Liu, Siyuan Lou, Jianxin Liao, and Hailin Feng. 2024. Dynamic and Static Representation Learning Network for Recommendation. *IEEE Trans. Neural Networks Learn. Syst.* 35, 1 (2024), 831–841.

[26] Yong Liu, Susen Yang, Yinan Zhang, Chunyan Miao, Zaiqing Nie, and Juyong Zhang. 2023. Learning hierarchical review graph representations for recommendation. *IEEE Trans. Knowl. Data Eng.* 35, 1 (2023), 658–671.

[27] Steffen Rendle. 2010. Factorization Machines. In *Proceedings of the 10th ICDM, Sydney, Australia, 14-17 December 2010.* IEEE Computer Society, 995–1000.

[28] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In *Proceedings of the NerrIPS 2007, Vancouver, British Columbia, Canada, December 3-6, 2007.* Curran Associates, Inc., 1257–1264.

[29] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Proceedings of the 11th ACM conference on recommender systems.* 297–305.

[30] Ana Salwa Shafie, Nurfadhlina Mohd Sharef, Masrah Azrifah Azmi Murad, and Azreen Azman. 2018. Aspect Extraction Performance with POS Tag Pattern of Dependency Relation in Aspect-based Sentiment Analysis. In *2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP).* 1–6.

[31] Liye Shi, Wen Wu, Wang Guo, Wenxin Hu, Jiayi Chen, Wei Zheng, and Liang He. 2022. SENGR: Sentiment-Enhanced Neural Graph Recommender. *Inf. Sci.* 589 (2022), 655–669.

[32] Liye Shi, Wen Wu, Wenxin Hu, Jie Zhou, Jiayi Chen, Wei Zheng, and Liang He. 2022. DualGCN: An Aspect-Aware Dual Graph Convolutional Network for review-based recommender. *Knowl. Based Syst.* 242 (2022), 108359.

[33] Jie Shuai, Kun Zhang, Le Wu, Peijie Sun, Richang Hong, Meng Wang, and Yong Li. 2022. A Review-aware Graph Contrastive Learning Framework for Recommendation. In *Proceedings of the 45th SIGIR, Madrid, Spain, July 11 -15, 2022.* ACM, 1283–1293. https://doi.org/10.1145/3477495.3531927

[34] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph Attention Networks. *CoRR* abs/1710.10903 (2017).

[35] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD, San Diego, CA, USA, August 21-24, 2011.* 448–456.

[36] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD, Sydney, NSW, Australia, August 10-13, 2015.* 1235–1244.

[37] Jiahui Wen, Jingwei Ma, Hongkui Tu, Mingyang Zhong, Guangda Zhang, Wei Yin, and Jian Fang. 2020. Hierarchical text interaction for rating prediction. *Knowledge-Based Systems* 206 (2020), 106344.

[38] Libing Wu, Cong Quan, Chenliang Li, Qian Wang, Bolong Zheng, and Xiangyang Luo. 2019. A context-aware user-item representation learning for item recommendation. *ACM Transactions on Information Systems (TOIS)* 37, 2 (2019), 1–29.

[39] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural Networks Learn. Syst.* 32, 1 (2021), 4–24.

[40] Lu Xu, Yew Ken Chia, and Lidong Bing. 2021. Learning Span-Level Interactions for Aspect Sentiment Triplet Extraction. In *Proceedings of the ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021.* 4755–4766.

[41] Chenyan Zhang, Shan Xue, Jing Li, Jia Wu, Bo Du, Donghua Liu, and Jun Chang. 2023. Multi-Aspect enhanced Graph Neural Networks for recommendation. *Neural Networks* 157 (2023), 90–102.

[42] Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the 10th ACM WSDM, Cambridge, United Kingdom, February 6-10, 2017.* 425–434.

[43] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. 2006. Learning with Hypergraphs: Clustering, Classification, and Embedding. In *Proceedings of the 20th NIPS, Vancouver, British Columbia, Canada, December 4-7, 2006.* MIT Press, 1601–1608.

A THE DETAILS OF ASPECT HYPERGRAPH CONSTRUCTION

Although Section 4.1 is self-explanatory, we would like to explain more about the AS-pair generation process in this section.

A.1 Rule-based extraction

The rule-based extraction method considers three dependency relations[3] sequentially used to extract candidate aspect-sentiment pairs, including *amod*, *nsubj+acomp*, and *dobj*, shown in Table 9.

On the one hand, some nouns directly describe the properties of items. They are considered potential aspects and are modified by adjectives with two types of dependency relations, *amod* and *nsubj+acomp*. The pairs of nouns and the modifying adjectives compose the AS-pair candidates. For instance, in the sentence *Amazing sound and quality, all in one headest*, the adjective *amazing* and the nouns *sound* compose a *amod* relationship. The nouns *sound* and *quality* are two aspects of an item, and the user thinks these aspects of the item are *amazing*, which is a positive sentiment. Thus, we extract adjectives as sentiment words and the related nouns as aspects. Sometimes, users tend to comment an aspect with a complete structure, such as *Quality is superior and comfort is excellent* , where *quality* is the subject and *superior* is the object. The dependency relation *nsubj+acomp* is suitable for this situation. On the other hand, the predicate and the object in a sentence describe the function of items. Thus, the combinations of predicates and objects are considered potential aspects by *dobj* dependency relation. In a combination, the predicate is regarded as the sentiment as it is usually with users' emotions and the whole combination is regarded as the aspect. For example, in the sentence *If you're recording vocals this will eliminate the pops,* the verb *eliminate* and noun *pops* construct a *dobj* relation. From this sentence, we know that the mentioned device is used to filter pops, and the word *eliminate* includes the user's emotions.

A.2 Filtering

To improve the quality of aspects, we merge synonyms and filter out aspects that are low in frequency. After that, we judge the polarities from sentiment words.

- Synonyms merging. A user may use two words with similar meanings for a particular method. Therefore, it is necessary to merge words with the same meaning. Merging synonyms closes the relationship between users and items obtained through aspect-sentiment pairs modeling and reduces the learning complexity of subsequent models. We collect the synonyms for each aspect and regard the most frequent synonym as the new aspect.
- Low-frequently aspects filtering. Filtering by setting a threshold can filter out part of the noise. In this paper, we set the threshold $c_t = 10$ to filter out the noise pairs.
- Sentiment polarity extraction. To simplify the modeling complexity, we use a sentiment analysis tool to judge the sentiment polarity of sentiment words. Three kinds of positive emotions, neutral emotions, and negative emotions were extracted from sentiment words [4].

B ADDITIONAL EXPERIMENTS

This section exhibits additional content regarding the experiments, such as a detailed experimental setup, the instructions to reproduce the baselines and our model, supplemental experimental results, and another case study. We hope the critical content helps readers gain deeper insight into the performance of the proposed framework.

B.1 Baseline

We compare APH with three types of baselines. Traditional rating-based methods include:

- **PMF** [28] is the probabilistic matrix factorization model, which is a classical collaborative filtering-based rating prediction method.
- **SVD**++ [17] is a classic matrix factorization method that exploits both the user's explicit preferences on items and the influences of the user's historical items on the target item.

Review-based methods include:

- **CDL** [36] is a hierarchical Bayesian model that employs SDAE for learning features from the content information and collaborative filtering for modeling the rating behaviors.
- **DeepCoNN** (DCN) [42] contains two parallel networks, which focus on modeling the user behaviors and learning the item properties from the review data.
- **NARRE** [3] uses an attention mechanism to model the importance of reviews and a neural regression model with review-level explanations for rating prediction.
- **CARL** [38] is a context-aware representation learning model for rating prediction, which uses convolution operation and attention mechanism for review-based feature learning and factorization machine for modeling high-order feature interactions.
- **DAML** [21] employs CNN with local and mutual attention mechanisms to learn the review features and improve the interpretability of the recommendation model.
- **NRCA** [23] uses a review encoder to learn the review representation and a user/item encoder with a personalized attention mechanism to learn user/item representations from reviews.
- **DSRLN** [25] extracts static and dynamic user interests by stacking attention layers that deal with sequence features and attention encoding layers that deal with of user-item interaction.

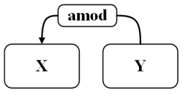
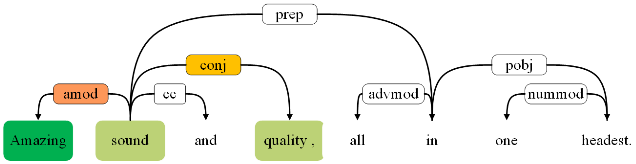
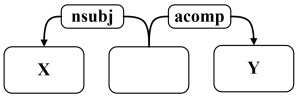
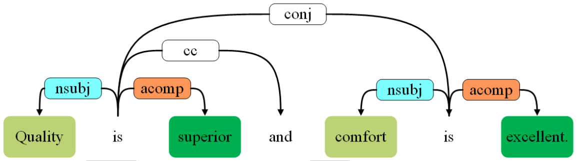
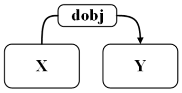
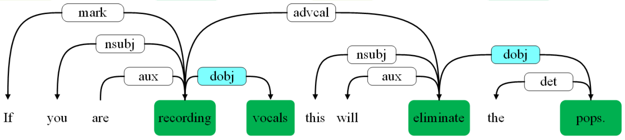
Aspect-based methods include:

- **ANR** [6] is an aspect-based neural recommendation model that learns aspect-based representations for the user and item by an attention-based module. Moreover, the co-attention mechanism is applied to the user and item importance at the aspect level.

[3] We use spaCy(https://spacy.io/) to extract the syntactic relations between the words.

[4] We use the Opinion Lexicon, which is available at https://www.cs.uic.edu/~lzhang3/programs/OpinionLexicon.html.

Table 9: Using dependency to extract aspect-sentiment pairs.

No.	Dependency		Example
AF1		**aspect: Y** **sentiment: X**	
AF2		**aspect: X** **sentiment: Y**	
AF3		**aspect: X_Y** **sentiment: X**	

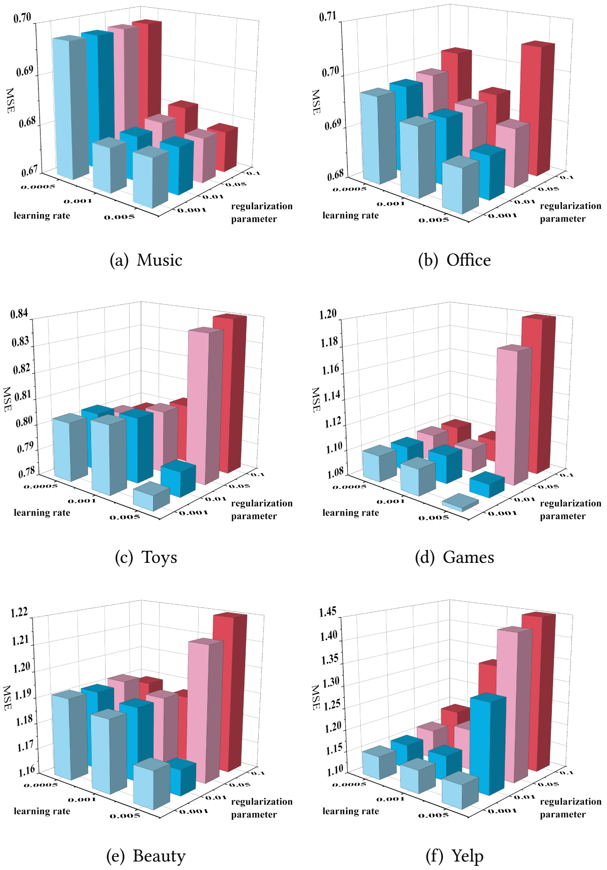
- **MA-GNNs** [41] predefines four aspects and constructs multiple aspect-aware user-item graphs, regarding the aspect-based sentiment as the edge. As it is trained by pairwise loss, we only compared it with NDCG.
- **RGNN** [26] builds a review graph for each user where nodes are words and edges are word orders. It uses a type-aware graph attention network to summarize graph information and a personalized graph pooling operator to capture important aspects.

B.2 Parameter Sensitivity Study

This subsection explores the effect of learning rate, regularization parameter, and embedding dimension.

B.2.1 The effect of learning rate and regularization parameter. We perform experiments to evaluate the sensitivity of APH to its hyperparameters. Following RGNN, the learning rate γ is varied in $[0.0005, 0.001, 0.005]$, and the regularization parameter λ is varied in $[0.001, 0.01, 0.05, 0.1]$. The experiment results are demonstrated in Figure 5, which shows the impact of hyperparameters γ and λ on six datasets. We can see that APH likes a small regularization parameter and a big learning rate. It achieves the best value on most datasets while $\gamma = 0.005$ and $\lambda = 0.001$.

B.2.2 The effect of embedding dimension. We perform experiments to evaluate the sensitivity of APH to its hyperparameters. The dimension of the semantic space d_1 and the hidden space of MLP d_2, are varied in $\{4, 8, 16, 32, 48, 64, 128.\}$ To reduce the computing cost, when we verify the impact of d_1, we set $d_2 = 8$, and that has the same setting for the verification of d_2. Figure 6 shows the MSE results on the Music dataset. We can see that APH achieves the best performance when $d_1 = 8$ and $d_2 = 8$. On the other five datasets, the MSE results show no significant difference in various dimensions settings. In APH, users and items are represented not only by their IDs' embedding but also by aggregating their aspect-sentiment graphs. Thus, APH has great power to represent users



(a) Music (b) Office

(c) Toys (d) Games

(e) Beauty (f) Yelp

Figure 5: Sparsity analysis of learning rate γ and the regularization parameter λ on six datasets.

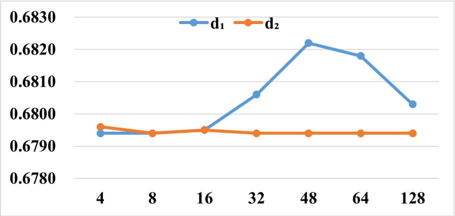
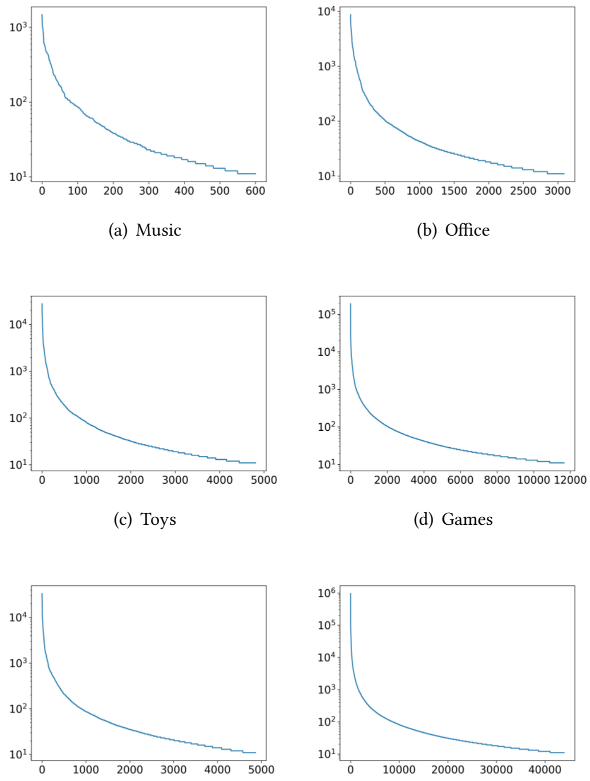


Figure 6: MSE of APH with various dimensions on Music dataset.



(a) Music

(b) Office

(c) Toys

(d) Games

(e) Beauty

(f) Yelp

Figure 7: Aspect distribution in six datasets.

and items by little dimension. For more experiments, please refer to the supplementary materials.

B.3 Results of extracted aspect

Our method aggregates explicit aspects to represent users and items and fuses them to make predictions. The aspects are extracted by an unsupervised method, discussed in section A. The number of aspects is smaller than that of items, and the number of quadruples is bigger than that of ratings. We also give the distribution of aspects in Figure 7. The aspect distributions of all datasets are long-tail distributions.