An Unsupervised Aspect-aware Recommendation Model with Explanation Text Generation

PEIJIE SUN, Hefei University of Technology, China
LE WU*, Hefei University of Technology, China
KUN ZHANG, Hefei University of Technology, China
YU SU, State Key Laboratory of Cognitive Intelligence, iFLYTEK, China
MENG WANG, Hefei University of Technology, China

Review based recommendation utilizes both users' rating records and the associated reviews for recommendation. Recently, with the rapid demand for explanations of recommendation results, reviews are used to train the encoder-decoder models for explanation text generation. As most of the reviews are general text without detailed evaluation, some researchers leveraged auxiliary information of users or items to enrich the generated explanation text. Nevertheless, the auxiliary data is not available in most scenarios and may suffer from data privacy problems. In this paper, we argue that the reviews contain abundant semantic information to express the users' feelings for various aspects of items, while these information are not fully explored in current explanation text generation task. To this end, we study how to generate more fine-grained explanation text in review based recommendation without any auxiliary data. Though the idea is simple, it is non-trivial since the aspect is hidden and unlabeled. Besides, it is also very challenging to inject aspect information for generating explanation text with noisy review input. To solve these challenges, we first leverage an advanced unsupervised neural aspect extraction model to learn the aspect-aware representation of each review sentence. Thus, users and items can be represented in the aspect space based on their historical associated reviews. After that, we detail how to better predict ratings and generate explanation text with the user and item representations in the aspect space. We further dynamically assign review sentences which contain larger proportion of aspect words with larger weights to control the text generation process, and jointly optimize rating prediction accuracy and explanation text generation quality with a multi-task learning framework. Finally, extensive experimental results on three real-world datasets demonstrate the superiority of our proposed model for both recommendation accuracy and explainability.

ACM Reference Format:
Peijie Sun, Le Wu*, Kun Zhang, Yu Su, and Meng Wang. 2018. An Unsupervised Aspect-aware Recommendation Model with Explanation Text Generation. *J. ACM* 37, 4, Article 111 (August 2018), 29 pages. https://doi.org/10.1145/1122445.1122456

Authors' addresses: Peijie Sun, Hefei University of Technology, No. 485, Danxia Road, Hefei, Anhui, China, sun.hfut@gmail.com; Le Wu*, Hefei University of Technology, No. 485, Danxia Road, Hefei, Anhui, China, lewu.ustc@gmail.com; Kun Zhang, Hefei University of Technology, No. 485, Danxia Road, Hefei, Anhui, China, zhang1028kun@gmail.com; Yu Su, State Key Laboratory of Cognitive Intelligence, iFLYTEK, No. 666, Wanjiang Road, Hefei, Anhui, China, yusu@iflytek.com; Meng Wang, Hefei University of Technology, No. 485, Danxia Road, Hefei, Anhui, China, eric.mengwang@gmail.com.

0004-5411/2018/8-ART111 $15.00
https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

With the broad applications on many online platforms, such as e-commerce website *Amazon*[1], and location based business website *Yelp*[2], many users like to express their preferences and write reviews to their consumed items. Review based recommendation has been emerged as a popular direction, which utilizes both users' rating behaviors and reviews for recommending items for users [46, 57]. Reviews could alleviate the data sparsity issue and help to improve recommendation accuracy. Moreover, reviews can explain why a user likes or dislikes an item. Thus reviews are usually used for constructing explainable recommendation, which is useful to help the system win users' trust and facilitate better recommendation conversion rate [41, 56].

Current works on review based recommendation can be grouped into two categories: accuracy oriented and explainability modeling. As embedding based recommendation models have shown state-of-the-art performance for accuracy modeling, works on the first category mainly leveraged content embedding of users and items from reviews, and fused the content embedding with collaborative filtering to enhance embedding representation ability of users and items [3, 29, 43, 44, 49, 53]. The second category considered providing explanation text for users when recommending items so that users can be easier to be persuaded. These kind of models borrow the success of the encoder-decoder based language generation techniques [32]. For example, in MRG [45] model, the target user-item ID embedding is encoded with a deep neural network first, and the recurrent neural network is used as a decoder to generate reviews with auxiliary multi-modal data and the encoded user-item ID embedding.

Despite the remarkable achievement that previous works have made, we argue that there still exist limitations in explanation text generation. Most of the users' reviews are general descriptions with little relevance to users' decision process [40, 55]. E.g., as shown in Fig.1, a user describes a restaurant with three sentences. Among them, the first two sentences "*What can I say that hasn't been said? I love this place.*" are general endorsements. Most of the current works input the user and item ID information without any specific semantic data, and rely on the review corpus with a large portion of general endorsements when training text generation models. As such, these models are likely to generate general explanation text, such as "*What a great place to eat*". To improve the generated explanation text quality with controlled specific information, a natural solution is to import auxiliary fine-grained information to enrich the input data. Researchers have proposed to import auxiliary data, such as item visual features[6, 26, 45], and knowledge graphs[8]. Some other works utilized external tools of Sentires [56] to extract feature related words from reviews, and then refined the raw reviews that have at least one feature word as the ground-truth [23]. However, the auxiliary information these models relied on may be not available. Besides, relying on external tools for feature extraction would let the performance of the proposed model be heavily influenced by the external tool.

In fact, users often utilize reviews to express their preferences for different aspects of items, such as taste and service. For example, in Fig. 1, a user *Lily* wrote her opinion about the restaurant she visited before in *Yelp.com*. We can obtain that *Lily* utilized "*My go to place for amazing pizza and pasta!*" to express her affirmation to an inferred aspect of *Italian Food*. From these highlighted phrases that *Lily* used to describe her feelings about the restaurant, we can infer that *Lily* concerns more about the *Italian Food* aspect of a restaurant. If we recommend a new restaurant to her, we should convince her to accept our recommendation from *Italian Food* aspect. In other words, the users' generated reviews contain expressive semantic information to explain her rating behavior, and could be mined to guide specific text generation for explainable recommendation.

[1]https://www.amazon.com
[2]https://www.yelp.com

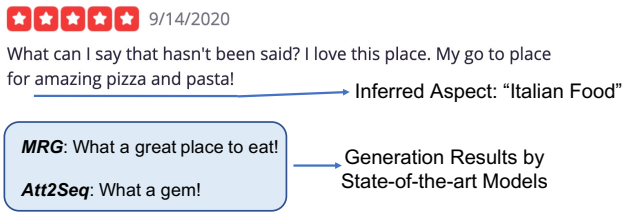

Fig. 1. A piece of restaurant review from Yelp.com, with the generated explanation text from two state-of-the-art models, ***MRG*** [45] and ***Att2Seq*** [16].

In this paper, we study the problem of how to provide aspect-guided explanation text in review based recommendation without any auxiliary data. To the best of our knowledge, ExpansionNet is one of the few attempts that incorporates aspect level information to control text generation, where the aspect representation is pre-trained from all users' review corpus [36]. However, the performance is unsatisfactory due to the following two reasons. First, ExpansionNet separates the processes of aspect extraction and text generation, making users and items not represented in the pretrained aspect space, thus leading to biases for aspect-guided text generation. Second, ExpansionNet neglects the correlation between the rating prediction and explanation text generation tasks for mutual improvement. Besides, simply utilizing the noisy reviews will lead to unsatisfied generated explanation text.

To this end, we focus on how to generate more fine-grained explanation text to persuade users without any auxiliary data. Though the idea is simple, it is not trivial. There are two main challenges. First, the aspect information in reviews is hidden and unlabeled. In other words, how to better utilize aspect information and represent user and item embeddings in the aspect space for recommendation is very challenging. Second, how to effectively utilize the aspect information to generate fine-grained explanation text with noisy user reviews is also full of challenges.

To tackle the above two challenges, we propose a novel multi-task learning framework named *Unsupervised Aspect-aware Explainable Review based Recommendation Model (*U-ARM*)*. It is designed for rating prediction and explanation text generation simultaneously. Specifically, we select a state-of-the-art unsupervised aspect extraction module (i.e., ABAE[19]) for better aspect extraction. By defining K latent aspects in an aspect space, *U-ARM* is capable of transforming each sentence of the review to the aspect space and calculating the aspect distribution of each review sentence. After that, users and items can be represented in the aspect space with their historical reviews. Then, the aspect-aware representations of users and items are injected into the rating prediction module and explanation text generation module. As users' original reviews are noisy and we aim to generate more persuasive explanation sentences, we further design an adaptive language generation loss that assigns larger weights to review sentences that have more proportion of aspect words. These three modules are trained in a multi-task learning manner, such that the aspect-based representations are shared among rating prediction and text generation for mutual enhancement of the two tasks. In summary, we inject the aspect information for representation learning, rating prediction, explanation text generation, and better review sentence importance learning in a unified framework. Finally, we conduct extensive experiments on three real-world datasets to verify the superiority and effectiveness of our proposed *U-ARM* for both rating prediction and explanation text generation.

2 RELATED WORK

2.1 Review based Rating Prediction

Given users' historical behavior, Collaborative Filtering (CF) models learn user and item embeddings from their historical data, and then predict each user's ratings to unconsumed items based on the similarity in the embedding space [5, 7, 15, 20, 38]. Despite the wide applicability, the performance is limited by the sparsity of the user-item rating matrix. As users often write reviews to express their feelings of items, review based recommendation utilizes both the reviews and users' rating behaviors to alleviate the sparsity issue. Earlier works have adopted topic models, i.e., Latent Dirichlet Allocation (LDA) [2], to extract the topic distributions of reviews. Then, the learned topic distributions are used as model regularization[27, 33, 46, 52], or part of the enriched hybrid user and item representations [43, 53]. With the huge success of deep learning in natural language processing, many researchers utilized state-of-the-art text embedding techniques in review based rating prediction models, including convolution neural networks(CNN) [57], recurrent neural networks (RNN) [30], and attention-based neural networks [39]. After that, similar to previous works, the learned review embeddings are also used as regularization [21, 30, 47, 54] or incorporated in CF for better user and item embedding modeling [3, 29, 39, 44, 49, 57].

As users express their feelings from various aspects in reviews, some researchers argued it is important to mine aspect information to improve recommendation accuracy. Some researchers proposed to leverage external aspect extraction tools to extract aspect words from reviews, and then aspect words and ratings are treated as the labels to help model optimization [48, 56]. Other researchers adopted graphical models for aspect modeling, with each aspect treated as an unobserved variable that guides observed reviews and ratings [9–11, 14, 18, 50]. The performance of these graphical models relies on the correlation assumption of hidden and observed variables, which needs to be predefined. Some researchers leveraged deep learning techniques to learn the aspect-based representations of users and items [12, 22]. However, the detailed correlation between the learned representation and semantics of each aspect is not clear.

Our work is closely related to aspect extraction [28]. Most previous aspect extraction models relied on extensions of LDA to treat a corpus as a mixture of topics (aspects). Recently, researchers proposed an **A**ttention-**B**ased **A**spect **E**xtraction model (ABAE) for unsupervised aspect extraction of a review sentence [19]. In this work, we leverage ABAE in our proposed model for aspect mining.

2.2 Explanation Text Generation Models

With the success of language generation models [32, 42], a choice for explainable recommendation is to generate explanation text. The language generation task usually has an encoder-decoder structure, in which the encoder part embeds the rich semantic information based on the detailed scenarios. E.g., machine translation task encodes the source sentence with a semantic vector [1], image and video captioning tasks focus on visual encoding in the encoder part [51]. For the explanation text generation task, we find there is no formal definition of the distinction between review generation and explanation text generation. According the related work, we summarize these related works into three categories.

First, researchers proposed to generate personalized text based on users' rating records and reviews. These models borrow the vanilla encoder-decoder architecture for text generation. Some researchers called the models as review generation [16, 35], while others treated it as explanation text generation [13, 31]. To generate explanation text based on the encoder-decoder structure, a natural choice is to send the user and item ID embedding as the input to the encoder for ID embedding, and let the RNN based decoder generate words one by one [13, 16, 24, 35]. In these models, the user and item embeddings are pretrained from the user-item rating matrix, or are

jointly learned with both rating prediction and the language generation optimization goal. These generation models are trained on review corpus. As a large portion of reviews are composed of general endorsements, most of the generated text is general without specification for the decision making process.

In the second category, researchers proposed to leverage auxiliary semantic information to generate more fine-grained explanation text. The auxiliary information includes visual features [6, 26, 45], knowledge graph [8], and so on. Most authors mapped the user and item to the same semantic space as the auxiliary first [6, 8, 26]. After that, the hidden state of the following text generation module is initialized by the representations of both user and item. And different attention mechanisms are utilized to control the text generation process [6, 26]. Most models in this category are called explanation generation, except the works that utilized the visual information [45].

As the auxiliary data is not always available, in the third category, researchers attempted to extract useful information from reviews or refine the noisy review datasets [4, 23, 34, 36]. For example, some researchers first manually annotated the justifications from reviews and then trained a classifier to classify whether a review sentence can be treated as justification [34]. The collected justifications are treated as the ground-truth. As human annotation is expensive, utilizing the external toolkit Sentires [56] to extract aspect words from the reviews is also a choice [4, 23]. Researchers treated the review sentences which contain at least one feature as ground truth [23]. Instead of manually constructing a new review dataset, researchers utilized auto-denoising mechanism to control the text generation process by assigning different weights to different review sentences [4]. Specifically, the review sentence which contains more proportion of feature words are assigned to larger weight, when calculating the text generation loss. These works are termed as explanation text generation in these original works [4, 23, 34]. Researchers also utilized external aspect extraction models such as [19] to extract aspect words first, and then proposed ExpansionNet to guide text generation with precompuated aspect information [36]. However, ExpansionNet is called as a review generation task.

To the best of our knowledge, ExpansionNet is one of the few attempts that considered how to generate specific aspect based review without any auxiliary data [36]. ExpansionNet is composed of two separate steps: the first is unsupervised aspect extraction [19]. After that, users and items are associated with aspect-aware representations to control the text generation. We differ from this model as follows. First and foremost, we treat the aspect space as a bridge for both user rating prediction and explanation text generation, and we could jointly perform these two tasks in a unified framework. In comparison, ExpansionNet treats the two tasks separately. Second, due to the separation of two tasks in ExpansionNet, this model could not associate user and item aspect-aware embedding vectors with detailed semantic entities in the latent aspect space, leading to inferior text generation performance. Third, ExpansionNet directly treats the noisy user reviews as review generation ground-truth, while our proposed model could adaptively learn better informative review for model training. In summary, we inject the aspect information for representation learning, rating prediction, explanation text generation, and better review sentence importance learning in a unified framework.

3 PRELIMINARY AND PROBLEM FORMULATION

In this section, we first introduce the state-of-the-art unsupervised aspect extraction model: ABAE [19], which is closely related to our work. Then, we give the problem formulation of review based recommendation.

3.1 Preliminary

As mentioned in Section 1, reviews contain abundant aspect information, which is helpful for generating fine-grained explanation text. However, the aspect information in reviews is hidden and

unlabeled, leading supervised methods inappropriate for aspect extraction. To this end, we intend to use unsupervised methods to obtain the aspect information from reviews. Since our focus is how to utilize aspect information to enrich both the rating prediction and explanation text generation processes, we select the state-of-the-art unsupervised aspect extraction model: ABAE [19] as the extraction model.

Formally, suppose there are K hidden aspects, given an input sentence $\mathbf{s} = (w^0, w^1, ..., w^t, ...w^T)$, ABAE model is capable of generating the aspect distribution $\phi^s \in \mathbb{R}^{K\times 1}$ of this sentence, which can be formulated as follows:

$$\phi^s = ABAE(\mathbf{s}), \tag{1}$$

where the k^{th} dimension of ϕ^s denotes the importance of the hidden k^{th} aspect.

Fig. 2(b) illustrates the architecture of ABAE. To be specific, the input sentence $\mathbf{s}$ is first embedded with a pre-trained word embedding matrix $\mathbf{E} \in \mathbb{R}^{V\times D}$, where V is the size of the vocabulary and D denotes the dimension of word representations. For example, we can get the embedding $\mathbf{e}^t$ of the t^{th} word w^t by indexing the matrix $\mathbf{E}$ with its index. Since not all words in sentence $\mathbf{s}$ have relations with aspects, the attention mechanism is employed to select the most relevant words. Thus, the sentence representation with the consideration of aspects can be modeled with the following equation:

$$\alpha^t = \mathbf{e}^t \mathbf{M}_a (\mathbf{y}^s)^\top, \quad \mathbf{z} = \sum_{t=1}^{T} \alpha^t (\mathbf{e}^t)^\top, \tag{2}$$

where $\mathbf{y}^s$ is the average of all word representations in sentence $\mathbf{s}$. $\mathbf{M}_a \in \mathbb{R}^{D\times D}$ represents the trainable parameters of attention mechanism. α^t denotes the importance of word w^t for the sentence. $\mathbf{z}$ is the sentence representation with the consideration of aspect information. With the help of attention mechanism, $\mathbf{z}$ can focus on words that are correlated to aspect distribution and represent sentence semantics in the aspect space.

Based on the aspect-aware sentence representation $\mathbf{z}$, the aspect distribution of the input sentence $\mathbf{s}$ can be calculated with the following equation:

$$\phi^s = Softmax(\mathbf{W}_\phi \mathbf{z} + \mathbf{b}_\phi), \tag{3}$$

where $\{\mathbf{W}_\phi \in \mathbb{R}^{K\times D}, \mathbf{b}_\phi \in \mathbb{R}^{K\times 1}\}$ are the trainable parameters. K is the pre-defined number of aspects. $Softmax$ function is utilized to calculate the proportion of each aspect in the input sentence. ϕ^s is the aspect distribution that will be used in our proposed model.

Next, ABAE defines an aspect matrix $\mathbf{K} \in \mathbb{R}^{D\times K}$ as model parameters to represent the K aspects information, with each aspect $\mathbf{K}_k$ is denoted as the k-th column of $\mathbf{K}$. The aspect matrix can be regarded as K points in the word embedding space. Based on the aspect distribution ϕ^s of sentence $\mathbf{s}$ and aspect matrix $\mathbf{K}$, ABAE can reconstruct the sentence embedding from its corresponding aspect distribution ϕ^s, which can be formulated as follows:

$$\mathbf{r}^s = \mathbf{K}\phi^s, \tag{4}$$

where $\mathbf{r}^s$ denotes the reconstructed sentence representation. To this end, ABAE can extract aspect information and generate aspect distribution of input sentences by minimizing the distance between sentence representation $\mathbf{z}$ and reconstructed sentence representation $\mathbf{r}^s$ in an unsupervised manner. The loss function and training details of ABAE will be detailed in Section 4.4.

According to Eq.(2) and Eq.(3), ABAE leverages attention mechanism to select the aspect-relevant words for aspect embedding. Each aspect is represented in the same word embedding space and can be treated as a cluster that aggregates similar aspect related words. Therefore, the learned hidden

aspects are naturally correlated to words. Based on the extracted aspect distribution, we are capable of representing users and items in the aspect space with their review text in a convenient way.

3.2 Problem Formulation

In a review based recommendation, the training set $\mathcal{X}$ consist of tuples (a, i, r_{ai}, C_{ai}). For each tuple, a and i denote the user ID and item ID. All users form the user set $\mathcal{U}$, $|\mathcal{U}|$=M, and all items form the item set $\mathcal{I}$, $|\mathcal{I}|$=N. r_{ai} and C_{ai} denote the rating and review of user a to item i. C_{ai} can be separated into several sentences $(s_{ai}^0, s_{ai}^1, ..., s_{ai}^{|C_{ai}|})$. And each sentence s_{ai}^l can be represented by a word sequence $(w^0, w^1, ..., w^t, ...w^T)$, where T is the length of s_{ai}^l and w^t denotes the t^{th} word in sentence s_{ai}^l. Then, each user a's review set is $C_a = ([C_{aj}], \forall j \in \mathcal{I}, r_{aj} > 0)$, and each item i's review set is $C_i = ([C_{bi}], \forall b \in \mathcal{U}, r_{bi} > 0)$. All the words which appear in all reviews form the vocabulary set $\mathcal{V}$. For any user-item pair (a, i), the target of review based recommendation with explanation text generation is to predict the corresponding rating and generate the corresponding explanation text based on the reviews set of C_a and C_i.

4 *UNSUPERVISED ASPECT-AWARE EXPLAINABLE REVIEW BASED RECOMMENDATION MODEL (U-ARM)*

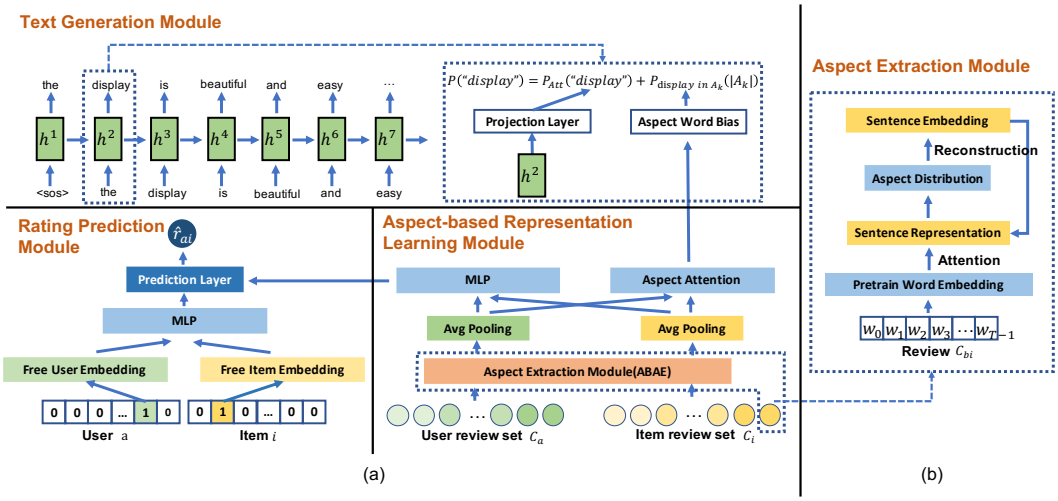

Fig. 2. (a) The overall architecture of our proposed model U-ARM. (b) The architecture of the unsupervised aspect extraction module ABAE.

Fig. 2 illustrates the overall architecture of our proposed *U-ARM*. In the following subsections, we will introduce the technical details of *U-ARM* from the following parts. First, we will describe how to learn the user and item aspect-based representations based on the outputs of the aspect extraction model. Then, we will introduce how to predict the corresponding rating and generate the explanation text for the user-item pair (a, i) based on their aspect-aware representations.

4.1 Aspect-based Representation Learning

In this module, we will introduce how to learn the aspect-aware representations of the users and items. As mentioned in Section 1, aspect information helps analyze the user preference from a detailed perspective. Therefore, it is natural to map the aspect-aware user and item representations

into the same aspect semantic space, which can avoid the biases for aspect guided text generation and support the analysis of the interaction between rating prediction and explanation text generation in review based recommendation.

Specifically, each user or item is correlated to a unique review set. We can generate the aspect distribution of each sentence in the review set. Then, a pooling operation is utilized to output the aspect-aware user and item representations in the same aspect space with their historical reviews. Taking the user-item pair (a, i) as an example, user a has a corresponding review set C_a, each review $C_{aj} \in C_a$ contains a set of sentences $(s_{aj}^0, s_{aj}^1, ..., s_{aj}^{|C_{aj}|})$, with $|C_{aj}|$ denotes the number of the sentences in review C_{aj}. The aspect distribution of each sentence can be calculated with the aspect extraction model $\mathcal{F}$:

$$\phi^{s_{aj}^l} = \mathcal{F}(s_{aj}^l), \tag{5}$$

where s_{aj}^l denotes the l^{th} sentence in the review C_{aj}. After getting the aspect distribution of each sentence, we employ the average pooling operation to map aspect-aware representation of each user into the aspect space. This process can be formulated as follows:

$$\phi_a = \frac{1}{\sum_{C_{aj} \in C_a} |C_{aj}|} \sum_{C_{aj} \in C_a} \sum_{s_{aj} \in C_{aj}} \phi^{s_{aj}}, \tag{6}$$

$\phi_a \in \mathbb{R}^{D \times 1}$ is the aspect-aware representation of user a in the aspect space. Similarly, the same operation is applied to the review set C_i of item i to get the aspect-aware representation ϕ_i of item i in the same aspect space.

4.2 Rating Prediction Module

This module will present how to make full use of the user and item aspect-aware representations, and how to encode user and item information in both the aspect space and latent space for better rating prediction. According to last subsection, we know the aspect-aware user and item representations are capable of modeling users and items in the same space as word embeddings and latent aspects, such that users and items are associated with aspects and words in the same semantic space. Thus, it is natural to ask how to make full use of these user and item semantic representations. Besides, as the interaction between users and items is also important, we intend to encode user and item information in both the aspect space and latent space. Then, we integrate them for better rating prediction.

Encoder. As the aspect-aware representations of users and items are learned from their corresponding reviews, they are capable of enriching the rating prediction module for better prediction. We first leverage a MLP to merge the aspect-aware representations of users and items. For each user-item pair (a, i), this process can be formulated as follows:

$$\mathbf{h}_{ai}^{\phi} = MLP_1([\phi_a, \phi_i]), \tag{7}$$

where $\mathbf{h}_{ai}^{\phi}$ denotes the encoded vector from the aspect space. $[\cdot, \cdot]$ is the concatenation operation. Here, MLP_1 represents the L_ϕ-layer multi-layer perceptron. For l^{th} layer of MLP_1, it can be formulated as follows:

$$\mathbf{h}_l^{\phi} = ReLU(\mathbf{W}_l^{\phi}\mathbf{h}_{l-1}^{\phi} + \mathbf{b}_l^{\phi}), \tag{8}$$

where $\{\mathbf{W}_l^{\phi}, \mathbf{b}_l^{\phi}\}$ are the trainable parameters. $ReLU$ is the non-linear Rectified Linear Unit activation function. $\mathbf{h}_{l-1}^{\phi}$ is the hidden states of the $(l-1)^{th}$ layer and $\mathbf{h}_0^{\phi} = [\phi_a, \phi_i]$. By stacking multiple

layers, *U-ARM* is able to make full of aspect-aware user and item representations for better rating prediction.

Apart from the aspect-aware representations, user and item representations in the collaborative latent space also play a crucial role. The collaborative signal can analyze the interaction between user and item and has achieved impressive performance. Concretely, we first set two free embedding matrices $\mathbf{P} \in \mathbb{R}^{M \times D_R}$ and $\mathbf{Q} \in \mathbb{R}^{N \times D_R}$ to represent users and items in the latent factor space, with $\mathbf{p}_a$ and $\mathbf{q}_i$ denote the latent factor vectors of the user a and item i, respectively.

Like the latent-factor models, we also utilize matrix factorization method to model the interaction between the user and item based on their attribute-based representations. To better coordinate with the encoding in aspect space and capture the complex interaction between the user and item, we employ another MLP to process the latent factor vectors of the user and item as follows:

$$\mathbf{h}_{ai} = MLP_2([\mathbf{p}_a, \mathbf{q}_i]), \tag{9}$$

where $\mathbf{h}_{ai}$ denotes the encoded vectors from the latent factor space. We have to note that MLP_2 has the similar structure as MLP_1.

Decoder for Rating Prediction. After getting the encoded vectors form the aspect space and latent factor space, it is natural to integrate them for final rating prediction. Here, we leverage linear transformer to predict the final rating of user a to item i, which can be formulated as follows:

$$\hat{r}_{ai} = \mathbf{W}_R(\mathbf{h}_{ai}^{\phi} + \mathbf{h}_{ai}) + \mathbf{b}_a + \mathbf{b}_i + \mu, \tag{10}$$

where $\mathbf{W}_R$ is the transfer matrix in the rating space. $\mathbf{b}_a$, $\mathbf{b}_i$ and μ are the biases of user, item and global average rating, respectively.

4.3 Explanation Text Generation Module

In this module, we aim to introduce how to leverage user and item aspect-aware representations to generate more fine-grained explanation text based on the real review $C_{ai} = (\mathbf{w}_{ai}^0, ..., \mathbf{w}_{ai}^t, ..., \mathbf{w}_{ai}^{|m|})$. Similar to many generation models, we employ an encoder-decoder structure for explanation text generation. To generate better specific text, we propose to leverage user and item aspect representations in the encoder part to learn their better aspect distribution, and let the decoder part to focus on specific aspects for more fine-grained explanation text generation. Next, we will give a detailed description about this generation module.

Encoder. The encoder part fuses the aspect-aware representations as well as the free latent representations of users and items. Let $\mathbf{P}^{'} \in \mathbb{R}^{M \times D_G}$ and $\mathbf{Q}^{'} \in \mathbb{R}^{N \times D_G}$ denote the free latent factors of users and items. A non-linear transformation is employed to encode the latent factor vectors and aspect-aware representations of the user and the item as:

$$\mathbf{u}_{ai} = tanh(\mathbf{W}_u[\mathbf{p}_a^{'}, \mathbf{q}_i^{'}] + \mathbf{b}_u), \quad \mathbf{v}_{ai} = tanh(\mathbf{W}_v[\phi_a, \phi_i] + \mathbf{b}_v), \tag{11}$$

where $\{\mathbf{W}_u \in \mathbb{R}^{D_H \times 2D_G}, \mathbf{W}_v \in \mathbb{R}^{D_H \times 2K}, \mathbf{b}_u \in \mathbb{R}^{D_H \times 1}, \mathbf{b}_v \in \mathbb{R}^{D_H \times 1}\}$ are trainable parameters, D_H is the dimension of the hidden state of the GRU structure in decoder. $tanh$ is the activation function.

Please note that, in the encoder part, the user and item aspect representations are shared with the aspect representations in the rating prediction module, as the user and item aspect representations are projected in the same hidden aspect space from the corresponding review set. Nevertheless, the user and item free latent matrices are different in the rating prediction and explanation text generation modules, as these two tasks rely on different user and item free embeddings. In practice, we also find setting different free latent matrices for these two tasks to achieve better results than sharing the same latent matrix in the experiments.

Decoder for Explanation Text Generation. Inspired by [17, 36], we utilize copy mechanism to inject aspect information in the explanation text generation process. For each predicted words

w_{ai}^t at step t, in order to inject the aspect information into the explanation text generation process, we argue that the predicted words are influenced by two kinds of factors: the previous sequence before step t, and the aspect distribution of the target user-item pair at step t. Thus, the predicted probability $p(w_{ai}^t)$ can be calculated with:

$$p(w_{ai}^t) = p_S(w_{ai}^t|w_{ai}^0, w_{ai}^1, ..., w_{ai}^{t-1}, \mathbf{p}_a', \mathbf{q}_i') + p_A(w_{ai}^t|t, \phi_a, \phi_i), \tag{12}$$

where the first term and the second term are the probabilities which are influenced by the previous sequence and the aspect distribution of the target user-item, respectively.

To model the influence of previous sequence, we simply employ GRU. Specifically, at training stage, we send the pre-trained embedding $\mathbf{e}_{ai}^t$ of the t^{th} word $\mathbf{w}_{ai}^t$ in the review C_{ai} to GRU, so that the hidden state $\mathbf{h}_{ai}^t$ can be updated for the t^{th} generation. This process can be formulated as follows:

$$\mathbf{h}_{ai}^0 = \mathbf{u}_{ai} + \mathbf{v}_{ai}, \quad \mathbf{h}_{ai}^t = GRU(\mathbf{h}_{ai}^{t-1}, \mathbf{e}_{ai}^t), \tag{13}$$

where $\mathbf{h}_{ai}^0$ is the initial hidden state of GRU. $\mathbf{h}_{ai}^{t-1}$ is the $(t-1)^{th}$ hidden state of GRU. To better exploit the encoded information, we employ attention mechanism to calculate the contribution of the user and item latent information for word prediction, as the latent factor representations focus on the interaction between user and item, both of their representations have influence on the final generation results. The process can be calculated as follows:

$$\begin{aligned} \alpha_a^t &= exp(tanh(\mathbf{W}_\alpha[\mathbf{p}_a^{'}; \mathbf{h}_{ai}^t] + b_\alpha))/Z, \\ \alpha_i^t &= exp(tanh(\mathbf{W}_\alpha[\mathbf{q}_i^{'}; \mathbf{h}_{ai}^t] + b_\alpha))/Z, \\ \mathbf{a}_{ai}^t &= \alpha_a^t \mathbf{p}_a^{'} + \alpha_i^t \mathbf{q}_i^{'}, \end{aligned} \tag{14}$$

where $\{\mathbf{W}_\alpha \in \mathbb{R}^{1\times(D_G+D_H)}, b_\alpha \in \mathbb{R}^1\}$ are trainable parameters. Z is the normalization term. And Z can be calculated with:

$$Z = exp(tanh(\mathbf{W}_\alpha[\mathbf{p}_a^{'}; \mathbf{h}_{ai}^t] + b_\alpha)) + exp(tanh(\mathbf{W}_\alpha[\mathbf{q}_i^{'}; \mathbf{h}_{ai}^t] + b_\alpha)). \tag{15}$$

After getting the weighted latent factor representation $\mathbf{a}_{ai}^t$ at step t, we concatenate it with the hidden state $\mathbf{h}_{ai}^t$ to predict the probability of the t^{th} word $\mathbf{w}_{ai}^t$ in the generated text:

$$p_S(w_{ai}^t) = tanh(\mathbf{W}_V[\mathbf{a}_{ai}^t, \mathbf{h}_{ai}^t] + \mathbf{b}_V). \tag{16}$$

In order to calculate the probability $p_A(w_{ai}^t)$, first we calculate the aspect distribution $\mathbf{b}_{ai}^t$ of the user-item pair (a, i) at step t, by leveraging the attention mechanism to calculate the contribution of their aspect-aware representations:

$$\begin{aligned} \beta_a^t &= exp(tanh(\mathbf{W}_\beta[\phi_a; \mathbf{h}_{ai}^t] + b_\beta))/Z^{'} \\ \beta_i^t &= exp(tanh(\mathbf{W}_\beta[\phi_i; \mathbf{h}_{ai}^t] + b_\beta))/Z^{'} \\ \mathbf{b}_{ai}^t &= \beta_a^t \phi_{\mathbf{a}} + \beta_i^t \phi_{\mathbf{i}}, \end{aligned} \tag{17}$$

where $\{\mathbf{W}_\beta \in \mathbb{R}^{1\times(K+D_H)}, b_\beta \in \mathbb{R}^1\}$ are trainable parameters. $Z^{'}$ is the normalization term. $\mathbf{b}_{ai}^t$ represents the distribution over K aspects of the user-item pair. With the learned $\mathbf{b}_{ai}^t$, the weights of each aspect in this user-item pair can be quantified. For each word w_{ai}^t, we can calculate its corresponding predicted probability $p_A(w_{ai}^t)$ with:

$$p_A(w_{ai}^t) = \mathbf{b}_{ai}^t \cdot \mathbb{1}_{w_{ai}^t \in \mathcal{A}_k}, \tag{18}$$

where $\mathcal{A}_k$ denotes all the words that associate with the k^{th} aspect. $\mathbb{1}_{w_k \in \mathcal{A}_k}$ is a binary value. If the word w_{ai}^t belongs to $\mathcal{A}_k$, $\mathbb{1}_{w_{ai}^t \in \mathcal{A}_k}$=1 otherwise it equals 0. Thus this equation means for any

candidate word w_{ai}^t that associate with the k^{th} aspect, its predicted probability is the value of the k^{th} element of the vector $\mathbf{b}_{ai}^t$.

Specifically, with the aspect matrix $\mathbf{K}$ defined in Eq.(4), each aspect $\mathbf{K}_A^k$ is represented in the same semantic space as words. Therefore, for each aspect k, we could construct an aspect word list $\mathcal{A}_k$ by selecting top-100 words that are most similar to $\mathbf{K}_A^k$ to form the vocabulary $\mathcal{V}_A$. In other words, each word in $\mathcal{A}_k$ has a close semantic relationship with this aspect in the inferred aspect space $\mathbf{A}$.

Finally, we combine Eq.(16) and Eq.(18) for text generation process as:

$$p(w_{ai}^t) = \begin{cases} tanh(\mathbf{W}_V[\mathbf{a}_{ai}^t, \mathbf{h}_{ai}^t] + \mathbf{b}_V) & \text{for} \quad w_{ai}^t \in \mathcal{V} - \mathcal{V}_A \\ tanh(\mathbf{W}_V[\mathbf{a}_{ai}^t, \mathbf{h}_{ai}^t] + \mathbf{b}_V) + \mathbf{b}_{ai}^t \cdot \mathbb{1}_{w_{ai}^t \in \mathcal{A}_k} & \text{for} \quad w_{ai}^t \in \mathcal{V}_A \end{cases}, \tag{19}$$

In the above text generation process, for the general words which belong to $\mathcal{V} - \mathcal{V}_A$, their predicted probabilities are only influenced by the previous sequence before step t. And for the aspect words which belong to $\mathcal{V}_A$, they are not only influenced by the previous sequence before step t but also the aspect distribution of the target user-item pair at step t. Thus we can judge at step t, whether the aspect words should be predicted, and which aspect words should be predicted. Based on this operation, *U-ARM* is capable of integrating the ID latent factor representations and aspect-aware representations of the user and item effectively, and generating fine-grained explanation text accurately.

4.4 Joint Model Training and Inference

In this section, we will present the joint model training and inference method of our proposed *U-ARM*.

Loss Function. The loss function is composed of three parts: the aspect extraction based loss, the rating prediction loss, and the explanation text generation loss.

As mentioned in Section 4, we employ aspect extraction module in an unsupervised manner. We select the margin based ranking loss. Specifically, both the reconstructed sentence representation $\mathbf{r}^s$ and weighed sentence representation $\mathbf{z}$ focus on the aspect information. We intend to minimize their distance as follows:

$$\mathcal{L}_A = max(0, 1 - < \mathbf{r}^s, \mathbf{z} > + < \mathbf{r}^s, \mathbf{z}^{'} >), \tag{20}$$

where $\mathbf{z}^{'}$ denotes the negative sentence representation, which we sample from the representations of other review sentences. We use the inner product $<,>$ to measure the similarity. Besides, the extracted aspects are encouraged to be different from each other, which can represent the aspect information comprehensively. In other words, any two columns of the matrix $\mathbf{K}$ are better to be orthogonal. Thus, we leverage the following equation to regularize the learning process of the learned aspect representation as:

$$\mathcal{L}_A^{reg} = (\mathbf{K}^\top \mathbf{K} - \mathbf{I})^2. \tag{21}$$

To this end, the loss function of unsupervised aspect extraction module is presented as follows:

$$\mathcal{L}_A^O = \lambda_1 \mathcal{L}_A + \lambda_2 \mathcal{L}_A^{reg}, \tag{22}$$

where λ_1 and λ_2 are the weights to balance the importance of different losses, which are defined before model training.

For rating prediction module, we select the Mean Square Error (MSE) as the loss function in the following format:

$$\mathcal{L}_R = \sum_{(a,i) \in \mathcal{X}} (r_{ai} - \hat{r}_{ai})^2, \tag{23}$$

where $\mathcal{X}$ denotes the observed user-item pairs in the training set.

For explanation text generation module, given the real reviews, we utilize the Negative Log Likelihood(NLL) as the loss function:

$$\mathcal{L}_G = -\frac{1}{|\mathcal{X}|} \sum_{(a,i)\in\mathcal{X}} \sum_{w_{ai}^t \in C_{ai}} \log\left(\text{softmax}\left(p\left(w_{ai}^t\right)\right)\right). \tag{24}$$

Alternatively, as not all content of the reviews are of explainable purpose, we have injected the learned aspects to adaptively learn review importance for better explanation text generation. Inspired by [4], by assigning each review sentence a weight according to the proportion of the aspect words of it, the explanation text generation process can be controlled. Thus, we calculate the importance score $\beta_{s_{ai}}$ of each review sentence s_{ai} with:

$$\beta_{s_{ai}} = \frac{N_{s_{ai}}}{|s_{ai}|}, \tag{25}$$

where $N_{s_{ai}}$ denotes the number of aspect words in the sentence s_{ai}, and $|s_{ai}|$ denotes the length of the sentence. The aspect words can be found in the aspect words set $\mathcal{V}_A$. Thus the loss function can be rewritten as:

$$\mathcal{L}_G = -\frac{1}{|\mathcal{X}|} \sum_{(a,i)\in\mathcal{X}} \sum_{s_{ai}\in C_{ai}} \beta_{s_{ai}} \sum_{w_{ai}^t \in s_{ai}} log(softmax(p(w_{ai}^t))). \tag{26}$$

Please note that the aspect extraction module is kept updated in the optimization process. Thus, the importance weight of each review sentence is also kept updated, as the aspect words are constructed with the aspect matrix $\mathbf{K}_A$.

To this end, the final loss function of our proposed *U-ARM* can be formulated as combining the unsupervised aspect extraction based loss function (Eq.(22)), the rating based loss function (Eq.(23)), and the language generation loss (Eq.(24) or Eq.(26)):

$$\mathcal{L} = \lambda_A \mathcal{L}_A^O + \lambda_R \mathcal{L}_R + \lambda_G \mathcal{L}_G, \tag{27}$$

where λ_A, λ_R and λ_G are the weights to balance the importance of different modules. λ_A controls the weight of the aspect extraction module, λ_R and λ_G are the weights for the two tasks of rating prediction and explanation text generation. The larger the weights, the more important the corresponding module in the model training process. Specifically, when $\lambda_A = 0$, i.e, the aspect based loss disappears in the final optimization function, our model turns into a classical model without any fine-grained aspect guided text generation. As there are two kinds of language generation loss, when Eq. (24) is treated as language generation loss, we use *U-ARM* to denote our proposed model. And when Eq. (26) is treated as the language generation loss, we use *U-ARM*-E to denote the enhanced versions of our proposed model.

4.5 Time Complexity

Comparing with the traditional models for rating prediction and explanation text generation, introducing the aspects will bring additional time complexity from three perspectives, i.e., aspect-aware representations learning, rating prediction, and explanation text generation. For any pair of user-item (a, i), the time complexity for the aspect-aware representations learning module is $O((L_a + L_i)(L(D^2 + 2D) + KD))$, where L_a, L_i denotes the review sentence number of the user a and item i, L denotes the maximum review sentence length, D denotes the word dimension and K denotes the number of aspects. In the rating prediction module, the time complexity which is brought by the aspect-aware representations can be $O(2DL_1 + L_1L_2 + L_2L_3 + 2L_3)$, if and only if the MLP in the rating prediction module is three-layers, and the layer dimensions of it are L_1, L_2,

and L_3 respectively. Moreover, in the explanation text generation module, the main operation is mapping the hidden state to the vocabulary space. As the dimension of the aspect-aware user and item representations is much smaller than the vocabulary size, it will not bring more significant time complexity. We can find that if an item or a user has many historical reviews, or if a sentence is too long, the time complexity in the aspect-aware representations learning module should be enormous. Thus, we control the maximum values of the L_a, L_i and L with p_1, p_2 and p_3, respectively. The p_1, p_2 and p_3 equal the values of the "Max Sentence Number of 85% Users", "Max Sentence Number of 85% Items", and "Max Sentence Length of 85% Sentences"in Table 1, respectively. More details can refer to subsection 5.2.

Table 1. Statistics for the three datasets.

Dataset	Amazon Video Games	Amazon Pet Supplies	Yelp(2020)
User Number	24,301	19,853	15,753
Item Number	10,672	8,510	16,110
Rating	231,780	157,836	162,004
Rating Density	0.089%	0.093%	0.064%
Max Sentence Number of All Users	627	117	96
Max Sentence Number of All Items	613	413	86
Max Sentence Length of All Sentences	571	372	242
Max Sentence Number of 85% Users	10	9	15
Max Sentence Number of 85% Items	27	22	13
Max Sentence Length of 85% Sentences	14	11	10
Vocabulary Size	35,902	16,602	22,228

5 EXPERIMENTS

In this section, we first introduce the datasets that we evaluate models on. Then, we describe the implementation details of our proposed *U-ARM* and *U-ARM*-E. Next, we present empirical results and give a detailed analysis of models on rating prediction and explanation text generation tasks, respectively.

5.1 Datasets.

We conduct experiments on three real world datasets, *Amazon Video Games*, *Amazon Pet Supplies*[3] and *Yelp(2020)*[4].

Amazon Video Games. This dataset is crawled from the online shopping website Amazon.com. It contains users' ratings, reviews, and some metadata from the customers to the products in "Video Games" category. Users' ratings range from 1 to 5, with larger ratings denote higher preferences.

Amazon Pet Supplies. This dataset is also crawled from the online shopping website Amazon.com. Products in this dataset belong to the "Pet Supplies" category. Values of the ratings also range from 1 to 5.

Yelp (2020). This dataset is provided by Yelp Inc. Yelp is a location based platform that provides user reviews and recommendations of the best restaurants, shopping, and so on. The dataset contains the ratings, reviews, and some metadata from the customers to various local business categories, such as "Restaurants", "Dentists", and "Bars". As the original dataset is too large, we only select the reviews of "Restaurants". In data pre-processing, we keep users and items with more than 10 and less than 100 training records.

[3]http://jmcauley.ucsd.edu/data/amazon/

[4]https://www.yelp.com/dataset

Table 2. Hyper-parameters configuration in *U-ARM* and *U-ARM*-E.

Hyper-parameter	**Value**
Aspect Number	$K = 15$
Aspect Embedding Size	$D = 200$
Free Embedding Size of $\mathbf{P}, \mathbf{Q}$	$D_R = 32$
Free Embedding Size of $\mathbf{P}', \mathbf{Q}'$	$D_G = 64$
Hidden Dimension of GRU in decoder	$D_H = 512$
MLP_1 Hidden Sizes	$[64, 128, 64, 32]$
MLP_2 Hidden Sizes	$[64, 128, 64, 32]$
Number of Layers in MLP_1	3
Number of Layers in MLP_2	3
Learning Rate	$lr = 0.001$
Batch Size	256
Learning Weight Decay	0.01

5.2 Data Processing.

We first randomly split the data into training set, validation set, and test set with 80%, 10%, 10%. Next, we employ two steps to process the data, including *Vocabulary Construction* and *Data Simplification*. The statistics of datasets after pre-processing are illustrated in Table 1.

1. Vocabulary Construction. We collect all reviews which appear in the training data and remove all stop words and punctuation. Then, we train a Word2Vec model based on the training review data with the gensim[5] toolkit, the parameters are set to (min_count=10, size=200, workers=12, iter=10). The vocabulary can be generated from the trained Word2Vec model and treated as the vocabulary for the rating prediction and explanation text generation modules.

2. Data Simplification. When preparing data for the aspect extraction module, the simplification process can be divided into two steps. First, we count the number of sentences in all the users' and items' corresponding reviews. Second, we randomly select at most p_1 sentences for each user and p_2 for each item. The p_1 and p_2 mean that 85% users at most have p_1 sentences and 85% items at most have p_2 sentences. And for each sentence, we also select at most p_3 words. p_3 means that 85% sentences have at most p_3 words. The p_1, p_2 and p_3 equal the values of the "Max Sentence Number of 85% Users", "Max Sentence Number of 85% Items", and "Max Sentence Length of 85% Sentences" in Table 1, respectively. We use the number 85 here because the authors adopted the same setting in their model implementation [57]. Please note that there is no mention of this setting in their paper, the details can be found in their implementation[6]. When preparing training data for the review generation module, we keep at most 30 words for each review.

5.3 Model Implementation

We tune the hyper-parameters on the validation set to achieve the best performance, and use early-stopping to select the best model. Since *U-ARM* has different hyper-parameter settings on different datasets, we list some common hyper-parameters in Table 2. Meanwhile, Adam optimizer is employed to optimize the model parameters. The entire model is implemented with PyTorch[7] and trained on Nvidia Tesla V100 GPU. The implementation of our proposed model can refer to this

[5]https://radimrehurek.com/gensim/index.html
[6]https://github.com/chenchongthu/DeepCoNN
[7]https://pytorch.org/

gitee repository[8]. And the hyper-parameter configures of all the comparative baseline models can refer to Table 3. We also release the implementations of these models in the same gitee repository.

To make our proposed model easier to train, we utilize the pre-training mechanism. Specifically, we first pretrain the rating prediction module with setting the $(\lambda_A, \lambda_R, \lambda_G)$ as $(1e-4, 1e+0, 1e-7)$ in Eq. (27). After that, we set the $(\lambda_A, \lambda_R, \lambda_G)$ as $(1e-4, 1e-3, 1e+0)$ and we load the pretrained rating prediction module to our proposed model.

In the inference process, for a new input user-item pair (a, i), we first collect all their related reviews, C_a and C_i. Then, we can get their aspect-based representations ϕ_a and ϕ_i with Equation (6). Next, we use these representations for rating prediction and explanation text generation.

Table 3. The hyper-parameter configures in all baseline models.

Model	**Hyper-parameter**	**Value**
PMF[38]	Latent Factor Dimension	32
NeuMF[20]	GMF Latent Dimension	32
	MLP Layers	64,128,64,32
DeepCoNN[57]	CNN Kernel Size	3
	CNN Filters Num	100
	FM Dimension	10
	Latent Factor Dimension	32
A3NCF[10]	Topics Num	15
	Num of Factors	30
	Activation Function	ReLU
GRU_LM[42]	Word Dimension	512
	Hidden Dimension	512
	Dropout	0.1
CF_GCN[35]	Encoder Dimension	200
	Decoder Dimension	512
	Hidden Dimension	512
	Dropout	0.1
MRG[45]	Latent Factor Dimension	32
	MLP Layers	64,128,64,32
	Word Dimension	512
	Hidden Dimension	512
	Dropout	0.1
Att2Seq[16]	Word Dimension	512
	Hidden Dimension	512
	Latent Factor Dimension	32
	Dropout	0.1
ExpansionNet[36]	Aspect Dimension	30
	Aspect Number	15
	Word Dimension	512
	Hidden Dimension	512
	Latent Factor Dimension	32
	Dropout	0.1

[8]https://gitee.com/PeijieSun/u-arm

5.4 Experiments on Rating Prediction

Baselines. We select six different baselines to compare with our proposed *U-ARM*, which can be grouped into three types: CF models, content-based models, and multi-task learning methods. The characteristics of these text generation baselines can refer to Table 4. The brief descriptions of these baselines are as follows:

- **AVG**: AVG predicts each user's rating is the average rating in the training data without any personalization. This simple baseline could be used to evaluate the improvement of recommender systems over non-personalized algorithms.
- **PMF** [38]: PMF is a classical recommendation model, which can model the linear collaborative interaction between users and items.
- **NeuMF** [20]: NeuMF boosts the performance of PMF by modeling both the simple and non-linear complex interaction between users and items.
- **DeepCoNN** [57]: DeepCoNN utilizes two CNN modules to learn the semantic-based representation of users and items, and predicts the rating based on Factorization Machine(FM).
- **A3NCF** [10]: A3NCF leverages LDA to extract the topic distribution of users and items, and designs an attention neural network to select the most valuable topics of users and items when predicting the rating.
- **CF-GCN** [35]: It is also a multi-task model. CF-GCN shares free user and item embedding of the two tasks, and utilizes the linear function to predict the rating for the target user-item pair.
- **MRG** [45]: MRG is a multi-task model for both review generation and rating prediction. It adopts an MLP module to predict rating from the user to the item.

Table 4. Characteristics of the baselines for rating prediction task. And characteristics of the baselines for explanation text generation task can be found in Table 6.

Model	Data Source		Aspect Modeling	Multi-task Learning
	Rating	Review		
PMF [38]	√	×	×	×
NeuMF [20]	√	×	×	×
DeepCoNN [57]	√	√	×	×
A3NCF [10]	√	√	√	×
CF_GCN [35]	√	√	×	√
MRG [45]	√	√	×	√
U-ARM	√	√	√	√

Metrics. Since it is a rating prediction task, we use the root mean square error (RMSE) as the evaluation metric, which is calculated as:

$$RMSE = \sqrt{\frac{1}{|\mathcal{X}|} \sum_{(a,i) \in \mathcal{X}} (r_{ai} - \hat{r}_{ai})^2}, \tag{28}$$

where $\hat{r}_{ai}$ and r_{ai} are the predicted rating and real rating.

Overall performance. Table 5 reports the rating prediction results of models on different datasets. We obtain the following three conclusions:

1. We observe that our proposed model outperforms other baselines in Amazon Video Games and Amazon Pet Supplies datasets. By taking the aspect information among reviews into consideration, *U-ARM* can represent users and items in the aspect space. Together with the

representations in latent space, users' ratings to items can be evaluated comprehensively. Meanwhile, the utilization of reviews can alleviate the data sparsity problem of these two datasets. Therefore, *U-ARM* can achieve the best performance.

2. Our proposed *U-ARM* ranks second on Yelp(2020) dataset. In fact, PMF shows the best performance on Yelp(2020), and CF-GCN that relies on PMF for rating prediction shows the same performance on this dataset. Models that built more complex users' personalized interests, e.g., non-linear user-item interaction with NeuMF, review content modeling with DeepCoNN, and fine-grained aspect modeling with *U-ARM*, do not perform as well as the simple PMF baseline. We guess a possible reason is that, for location based recommendation like Yelp, apart from the personalized preference, there are many local features that may influence users' choices, such as popularity, location and so on. These factors are not easy to mine from reviews and user-item behaviors.
3. When talking about the aspect information, *U-ARM* not only performs better than the aspect-based model A3NCF, but also outperforms the review based model DeepCoNN. This phenomenon demonstrates that properly modeling aspect information is also helpful for rating prediction. Besides, as the datasets are very sparse, complex deep learning models like NeuMF may perform worse than shallow models like PMF.

Table 5. Rating prediction performance with RMSE metric. $*$ means our proposed model can achieve significant improvement than all the baseline models with p<0.05 based on the Student's t-test. ** means our proposed model can achieve significant improvement than most baseline models with p<0.05 based on the Student's t-test. The bold font refers to the best model and the underline shows the model that ranks the second.

Model	Amazon Video Games	Amazon Pet Supplies	Yelp(2020)
AVG	1.2039	1.1722	1.2366
PMF	1.0520	<u>1.1044</u>	**1.1239**
NeuMF	<u>1.0435</u>	1.1058	1.1290
DeepCoNN	1.0453	1.1051	1.1319
A3NCF	1.0694	1.1417	1.1353
CF-GCN	1.0513	1.1148	**1.1239**
MRG	1.0478	1.1055	1.1271
U-ARM	**1.0393***	**1.1032***	<u>1.1258</u>**

5.5 Explanation Text Generation

Baselines. For this task, we select five state-of-the-art baselines to compare with our proposed model *U-ARM*, including GRU_LM, Att2Seq, MRG, CF_GCN, ExpansionNet. The characteristics of these text generation baselines can refer to Table 6. Following are brief descriptions of these baselines.

- **GRU_LM** [42]: GRU_LM model is a base language generation model without detailed user and item ID embedding modeling.
- **Att2Seq** [16]: Att2Seq is designed to encode both the ID information of the user and item, and the rating information for review text generation task.
- **ExpansionNet** [36]: ExpansionNet is a state-of-the-art model to generate reviews by injecting the aspect information. ExpansionNet first leveraged a prerained ABAE model to find aspect related words. Then, users and items are associated with aspect-aware representation to control the generation by attending to specific encoder information.

- **CF_GCN** [35]: It is also a multi-task model for both tasks. Different from MRG, it treats the linear interaction of the user and item as the input of the text generation module.
- **MRG** [45]: MRG is a multi-task model for both review generation and rating prediction. It models the complex interaction between users and items as the input of the following text generation module. In our work, as we don't have any multimodal data, we will simplify the text generation module with text input.

Table 6. Characteristics of the baselines for review generation task. The characteristics of the baselines for rating prediction task can be found in Table 4.

	Data Source		Aspect Modeling	Multi-task Learning
	Rating	Review		
GRU-LM [42]	×	√	×	×
Att2Seq [16]	√	√	×	×
ExpansionNet [36]	×	√	√	×
CF_GCN [35]	√	√	×	√
MRG [45]	√	√	×	√
U-ARM	√	√	√	√

Metrics. For language generation task, we employ *BLEU_1, BLEU_4, ROUGE_1_F* and *ROUGE_L_F* as the metrics [25, 37]. These metrics are calculated based on the overlapping content of the candidate generated reviews and the real reviews. Larger values of these metrics mean better performance. More specifically, BLEU is used to evaluate how many generated words appear in real reviews. ROUGE is used to evaluate how many real words appear in generated reviews. Besides, BLEU_1, BLEU_4 are calculated based on the unigram and 4-grams. ROUGE_1_F and ROUGE_L_F are calculated based on the unigram and n-grams, where n is the length longest co-occurring in sequence, and n is calculated automatically.

To our best knowledge, it is non-trivial to evaluate the performance of the explanation text generation models directly as there is no ground-truth. In this paper, since our proposed models aim to generate more fine-grained aspect-aware explanation text, general sentences in reviews should be removed for better evaluation, such as "What can I say that hasn't been said? I love this place." in Figure 1 in the previous version, which is not helpful to generate persuasive explanation text. However, this operation is time and labor consuming. In this paper, we evaluate the quality of the generated explanation text from two perspectives. First, to guarantee our proposed models can capture the interaction between the target user and item and generate reasonable explanation sentences, we utilize the BLEU_* and ROUGE_* metrics to automatically evaluate whether our proposed models can generate actual sentences or not. From Table 7 and Table 12 in the revised version, we can find our proposed models outperform almost all baseline models based on the BLEU_* and ROUGE_* metrics. Second, to evaluate the explainability of our proposed models, we conduct experiments based on three metrics, "Fluency", "Coherence", and "Persuasiveness". From Table 14, we can find that our proposed model also outperforms all baseline models. Therefore, based on the above experimental results, the explainability of our proposed models can be guaranteed.

Overall performance. We report the experimental results of all models on three datasets in Table 7. According to the results, two important observations are presented as follows:

1 Among all models,*U-ARM* achieves the best performance on all datasets with all metrics, followed by the baseline ExpansionNet. This phenomenon shows it is important to inject fine-grained information for review generation. Nevertheless, ExpansionNet separates the

process of aspect extraction and text generation. On the contrary, *U-ARM* integrates the aspect extraction and user-item encoding processes, and mutually enhance the two tasks of rating prediction and review text generation. Therefore, *U-ARM* not only avoids the biases for aspect-guided generation, but also selects the relevant aspect information for the generation process, which leads to the best performance on the explanation text generation task.

2 CF_GCN uses simple linear concatenation of user and item free embeddings, and MRG adopts the MLP structure for user and item embedding fusion. Though the structures of MRG and CF_GCN are very similar, we can observe from Table 7 that CF_GCN outperforms MRG. We speculate one possible reason is that they both leverage limited user behavior data to model the additional complex user-item free embedding interaction. Although the encoding module of our proposed model *U-ARM* is also complex, we represent the user and item in the aspect semantic space with their historical reviews. As the reviews contain more meaningful semantic information, the aspect space and the behavior space need to be carefully fused in the encoder part for better explanation text generation. Therefore, our encoding module, although complex, is able to achieve better performance.

Table 7. Explanation text generation: model performance on BLEU and ROUGE metrics on three real-world Datasets. $*$ means our proposed model can achieve significant improvement than all the baseline models with p<0.05 based on the Student's t-test. ** means our proposed model can achieve significant improvement than most baseline models with p<0.05 based on the Student's t-test. The bold font refers to the best model and the underline shows the model that ranks the second.

Model	Amazon Video Games			
	BLEU_1	BLEU_4	ROUGE_1_F	ROUGE_L_F
GRU-LM	0.2537	0.0066	0.2877	0.206
Att2Seq	0.2553	0.0087	0.2918	0.2058
ExpansionNet	0.2643	<u>0.0105</u>	<u>0.3053</u>	<u>0.2147</u>
CF-GCN	**0.2654**	0.0102	0.3021	0.2143
MRG	0.2581	0.0082	0.2951	0.2088
U-ARM	<u>0.2650</u>**	**0.0112***	**0.3081***	**0.2163***
Model	Amazon Pet Supplies			
	BLEU_1	BLEU_4	ROUGE_1_F	ROUGE_L_F
GRU-LM	0.2274	0.0041	0.2604	0.1907
Att2Seq	0.2395	0.0056	0.2735	0.1928
ExpansionNet	0.2546	<u>0.0088</u>	<u>0.2972</u>	<u>0.2055</u>
CF-GCN	<u>0.2559</u>	0.0078	0.2915	0.203
MRG	0.2402	0.0058	0.2753	0.1938
U-ARM	**0.2608***	**0.0089***	**0.3009***	**0.2082***
Model	Yelp(2020)			
	BLEU_1	BLEU_4	ROUGE_1_F	ROUGE_L_F
GRU-LM	0.2349	0.0063	0.2586	0.1946
Att2Seq	0.2314	0.0053	0.2525	0.1907
ExpansionNet	<u>0.2427</u>	<u>0.0070</u>	<u>0.2736</u>	0.1993
CF-GCN	0.2356	0.0063	0.2703	<u>0.1995</u>
MRG	0.2377	0.0062	0.2629	0.1934
U-ARM	**0.2446***	**0.0074***	**0.2755***	**0.2010***

Table 8. List of representative words for several inferred aspects.

Amazon Video Games	
Inferred Aspect Name	Aspect Words
Installation	configure, access, applied, mapped, hotkey, transmits, coordinate, vibrates
Elements	battlepacks, unlock, wealth, badges, capsules, bonuses, psionic, prestige
Game Name	rdr, cnc, rpg, genre, nmh, puzzler, gtas, wargaming
Time	tonight, june, august, december, july, january, shipped, screamed
Scene	explores, malgrave, switzerland, monumental, harbors, hellgate, oceanic, geographic
Amazon Pet Supplies	
Inferred Aspect Name	Aspect Words
Dog breeds	pug, doberman, westie, beagle, labradoodle, maltipoo, puggle, goldendoodle
Pet status	olds, itched, fracture, bratty, limp, lethargic, howled, steadily, luckily
Material	vinyl, plastic, canvas, rubberized, silicone, elastic, pvc, neoprene
Animal type	dogs, cats, ferrets, rats, weimaraners, parakeets, bunnies, turtles
Body size	small, large, big, tiny, giant, larger, shallow, wide
Yelp(2020)	
Inferred Aspect Name	Aspect Words
Service	rude, subpar, impatient, unhappy, pushy, flustered, upset, lacklustre
Taste	flavorful, tasty, vinegary, waterlogged, juicy, crispy, meaty, tender
Snacks	blackberry, marshmallow, sorbet, jellies, crackers, espressos, passionfruit, raisins
Location	hudson, columbus, matthews, mckenzie, santan, beaches, kennedy, braddock
Price	affordable, quality, inexpensive, inventive, upscale, modest, cheap, bargain

5.6 Aspect Modeling Performance

As mentioned in the previous sections, utilizing aspect information from reviews is the key characteristic of our proposed *U-ARM*, which could generate fine-grained explanation text to persuade users. To this end, we make extra exploration about aspect modeling to verify its effectiveness on explanation text generation and better demonstrate the superiority of our proposed *U-ARM*. In the following parts, we give a detailed analysis about aspect modeling from three directions: *The selected aspect words*, *The proportion of aspect words in generated text*, and *The case study of generated text*.

The selected aspect words. To show the extracted aspect information more intuitively, we list five inferred aspects and the associated representative words for each dataset. The aspect words are obtained as follows: we first train our proposed *U-ARM*. After that, for each learned aspect representation, we find the top similar words based on the cosine similarity between the aspect representation and the word representation. We list some aspects and the words that appear in each aspect in Table 8. As we do not have any ground truth labels of the real aspects in these datasets, we infer the most suitable aspects from the corresponding extracted words for easier understanding. Based on the results in Table 8, we can conclude that these aspects can better describe the user preference from different dimensions. For example, In pet stuff shopping, users concern more about that have a connection with the pets, such as *pet status, animal type, body size*, and so on. For the inferred aspect *pet status*, the closely related words, such as *olds*, *itched*, *fracture*, describe the possible healthy concerns with the current pet status. This aspect shows the health status of each pet, and can be used to better describe the particular effect for each possible health status. As such, these aspect words can help *U-ARM* to model the interaction between users and items in a detailed manner, which is in favor of predicting more accurate ratings of users to items. Moreover, these extracted words describe the specific attributes of items. With their help, *U-ARM* can generate a more fine-grained explanation text for convincing users. In other words, *U-ARM* has better performance on explanation text generation task.

The statistics of aspect words in the generated text. We have already shown that the performance of explanation text generation has been improved with language evaluation metrics. All these language evaluation metrics are designed to evaluate the overlap of the generated sentence and the real reviews provided by users. Since most users' reviews contain a large portion of general terms without specific product attributes for the explanation, it is still unclear whether the generated explanation text is more specific and can better persuade users based on their personalized needs. To this end, in this part, we intend to evaluate the specificity of the generated text. Since each hidden aspect and each word are mapped into the same semantic space, for each aspect, we also first calculated the top-100 nearest words of each aspect, and treated the nearest words as aspect words. For each dataset, we set the number of aspects K as 15. We report the statistics of unique aspect words in Table 9. In this table, "Number of Unique Aspect Words" of *GroundTruth* denotes the number of unique aspect words in the test data. "Percentage of Aspect Words" denotes the percentage of aspect word of the generated explanation text, and "Percentage of Common Aspect Words" represents the percentage of aspect words that appear in both the real review text and the corresponding generated explanation text simultaneously. These manually defined metrics are used to evaluate the usefulness of the generated text from various perspectives. Please note that, similar metrics are also used in previous works [4, 23]. In these closely related works, researchers proposed to use external toolkit Sentires [56] to pick features, and test whether these feature words appear in the generated text.

Based on the statistical results, we can observe that the generated explanation text of the Att2Seq model contains the most aspect words. For our proposed model *U-ARM*, there are merely about one-half of all the aspect words in the generated explanation text. Nevertheless, in the "Percentage of Common Aspect Words" column, our proposed *U-ARM* achieves the most similar results to the ground truth, which demonstrates that *U-ARM* makes full use of aspect words in the most appropriate way. After analyzing the model structure in detail, we obtain that Att2Seq model leverages the attention module to encode the user and item representations, so that it can better generate more diverse sentences with diverse aspect words. However, it treats the text generation as a single task, which cannot avoid the biases for aspect-guided text generation and cannot satisfy the personalized need of users. On the contrary, *U-ARM* treats the rating prediction and explanation text generation tasks as a joint optimization target. Besides, although the attention mechanism

Table 9. The statistics of the aspect words for all explanation text generation models. $*$ means our proposed model can achieve significant improvement than all the baseline models with p<0.05 based on the Student's t-test. ** means our proposed model can achieve significant improvement than most baseline models with p<0.05 based on the Student's t-test. The bold font refers to the best model and the underline shows the model that ranks the second.

Model	Amazon Video Games		
	Number of Unique Aspect Words	Percentage of Aspect Words	Percentage of Common Aspect Words
GroundTruth	*880*	*4.69%*	-
GRU_LM	283	0.44%	0.15%
Att2Seq	**639**	<u>1.91%</u>	<u>0.20%</u>
ExpansionNet	384	1.27%	0.19%
CF_GCN	476	1.19%	0.19%
MRG	<u>512</u>	1.43%	0.18%
U-ARM	393	**2.68%***	**0.26%***
Model	Amazon Pet Supplies		
	Number of Unique Aspect Words	Percentage of Aspect Words	Percentage of Common Aspect Words
GroundTruth	*1,040*	*6.26%*	-
GRU_LM	383	1.15%	0.19%
Att2Seq	**770**	<u>3.61%</u>	0.43%
ExpansionNet	492	3.08%	<u>0.55%</u>
CF_GCN	<u>618</u>	2.76%	0.43%
MRG	537	2.07%	0.30%
U-ARM	534	**4.04%***	**0.61%***
Model	Yelp(2020)		
	Number of Unique Aspect Words	Percentage of Aspect Words	Percentage of Common Aspect Words
GroundTruth	*960*	*4.13%*	-
GRU_LM	436	1.30%	0.20%
Att2Seq	**750**	**2.37%**	0.24%
ExpansionNet	<u>468</u>	1.44%	<u>0.24%</u>
CF_GCN	254	0.53%	0.20%
MRG	465	0.95%	0.20%
U-ARM	461	<u>1.96%</u>**	**0.25%***

is also used in ExpansionNet [36] and our proposed model, the diversity of the aspect words is constraint by the copy mechanism. Nonetheless, based on the copy mechanism, the words generated by our proposed model are more concentrated in those words that appear in the target user-item pairs. This is also the reason why our proposed model generates fewer unique aspect words than the Att2Seq model while achieving a higher "Percentage of Common Aspect Words" in most cases. Along this line, not only the biases for aspect-guided generation can be avoided, but also the most relevant aspect information can be selected for the generation process. To this end, our proposed *U-ARM* achieves a better performance.

The case study of generated text. In the previous part, we have demonstrated that the aspect information can help to describe the user preference in a detailed manner and enhance the model

Table 10. Generated reviews of several selected methods. Bold fonts are for emphasis.

Amazon Video Games	Real Review	I have been a long time **gta** player. I have been playing gta from the time the very **1st one** came out. I liked all the other **gta games**.
	Att2Seq	I have to say I love this **game**. It is a **sensational** game.
	ExpansionNet	I love the **game**. I have been playing it for a few hours and I have to say I was really **impressed**.
	MRG	I love this **game**. I love the **graphics** and the **gameplay**. The **graphics** are **amazing**.
	U-ARM	I have been a fan of the **gta series** since the **first one** came out. I was a **huge fan** of the **gta series**.
Amazon Pet Supplies	Real Review	Arrived in a **timely manner**. One of the first liquid supplements I bought for my **5 gallon nano tank**. Mainly had anubias and java fern.
	Att2Seq	Arrived **promptly**, and it 's ok. I have been using this product for years. It is very **stable**. I have been using it for about a week now.
	ExpansionNet	Arrived in a **timely manner** and the packaging isn't exactly as described. I have a **29 gallon tank** and I haven't had any issues with this product.
	MRG	Arrived **on time**, and I 'm happy with the outcome of the tank. I recommend this product. It 's a **good** product. I recomend it to anyone.
	U-ARM	Arrived in a **timely manner**. I have used it for a few days and it is very **effective**. I have used it for **my 29 gallon freshwater tank**.
Yelp(2020)	Real Review	What can i say that hasn't been said? I love this place. My go to place for **amazing** pizza and pasta!
	Att2Seq	What a gem !
	ExpansionNet	What a great place to get a great **italian food**. I love the **italian sausage**. This is the best italian restaurant in cleveland.
	MRG	What a **great** place to eat !
	U-ARM	What a **great find**! I love this place! The staff is **friendly** and the food is **delicious**. The italian sausage is a **must try**. The pizza is **delicious**.

ability of rating prediction and explanation text generation simultaneously. In this part, we randomly select some cases from three datasets, and list the generated explanation text of our proposed *U-ARM* and some baselines. Since GRU_LM model relies on the start word, and CF-GCN model is very similar to the MRG model, we do not list their results. Table 10 reports the corresponding results. The bold words denote the specific aspect information. Based on the results, we can observe that these models without aspect information utilization generate general explanation text, such as "*I love this game*, *What a gem!*". When considering the aspect information, models have better performance. For example, ExpansionNet can generate more reasonable explanation text, such as "*I have been playing it for a few hours*, *I love the Italian sausage*", to convince the users. However, its

Table 11. Ablation study of our proposed model *U-ARM* on Amazon Pet Supplies dataset. The metric RMSE is used to evaluate the performance of the proposed model for the rating prediction task. And the metrics BLEU and ROUGE are used to evaluate the performance of the model for the explanation text generation task.

U-ARM		RMSE	BLEU_1	BLEU_4	ROUGE_1_F	ROUGE_L_F
Different Number of Aspects	K=0	1.1045	0.2357	0.0061	0.2745	0.1929
	K=5	1.1036	0.2582	0.0086	0.2993	0.2069
	K=10	1.1034	0.2607	0.0088	0.2996	0.2078
	K=15	1.1032	0.2608	0.0089	0.3009	0.2082
	K=20	1.1034	0.2570	0.0088	0.2985	0.2065
Different Explanation Text Generation Modules	$\mathbf{0} \rightarrow \mathbf{h}^0_{ai}$;$\mathbf{h}^t_{ai} \rightarrow w^t_{ai}$	1.1029	0.2536	0.0065	0.2892	0.2033
	$[\mathbf{p}'_a, \mathbf{q}'_i] \rightarrow \mathbf{h}^0_{ai}$;$\mathbf{h}^t_{ai} \rightarrow w^t_{ai}$	1.1030	0.2534	0.0084	0.2970	0.2045
	$[\mathbf{p}'_a, \mathbf{q}'_i] \rightarrow \mathbf{h}^0_{ai}$; $[\mathbf{h}^t_{ai}, \mathbf{p}'_a, \mathbf{q}'_i] \rightarrow w^t_{ai}$	1.1029	0.2542	0.0088	0.2985	0.2053
	$[\mathbf{p}'_a, \mathbf{q}'_i, \phi_a, \phi_i] \rightarrow \mathbf{h}^0_{ai}$; $[\mathbf{h}^t_{ai}, \mathbf{p}'_a, \mathbf{q}'_i, \phi_a, \phi_i] \rightarrow w^t_{ai}$	1.1032	0.2608	0.0089	0.3009	0.2082
Shared Latent Embeddings ($\mathbf{P} = \mathbf{P}'$)		1.1039	0.2315	0.0049	0.2657	0.1925

performance is still incomparable with our proposed *U-ARM*. By taking the correlation between rating prediction and explanation text generation text into consideration, *U-ARM* can utilize the aspect information in a detailed and precise manner, so that it can generate more reasonable and fine-grained explanation text for users. For example, *U-ARM* can generate "*I was a huge fan of the **gta series**, The **italian sausage** is a **must try**. The **pizza** is delicious*"with the user concerned aspects. In a word, *U-ARM* does make full use of aspect information to generate more fine-grained explanation text for users.

5.7 Ablation Study

To better study the effectiveness of each component of our proposed model *U-ARM*, we treat Eq. (24) as the explanation text generation loss. And we test *the components with different numbers of aspects(K), different explanation text generation modules*, and *shared latent embeddings*. Please note that, when we test one component of our proposed model *U-ARM*, we keep the other components and parameter setting as the same of the ones we introduced before.

Different numbers of aspects(K). From the first four rows in Table 11, we can find the performance of our proposed model varies with the value of K. When K=15, our proposed model performs best. Therefore, we also assign 15 to K when conducting experiments. By the way, K=0 means there is no aspect extraction module in our proposed model. Thus the recommendation accuracy decreases rapidly.

Different explanation text generation modules. In Table 11, "$\mathbf{0} \rightarrow \mathbf{h}^0_{ai}$;$\mathbf{h}^t_{ai} \rightarrow w^t_{ai}$ " means the hidden state in Eq. (13) is initialized by zero vector, and the predicted probability of the word is just calculated by Eq. (16) without the representation $\mathbf{a}^t_{ai}$. "$[\mathbf{p}'_a, \mathbf{q}'_i] \rightarrow \mathbf{h}^0_{ai}$;$\mathbf{h}^t_{ai} \rightarrow w^t_{ai}$ " means the hidden state is only initialized with the latent factor vector $\mathbf{u}_{ai}$ of the user-item pair in Eq. (13), and the predicted probability of the word is just calculated by Eq. (16) without the representation $\mathbf{a}^t_{ai}$. "$[\mathbf{p}'_a, \mathbf{q}'_i] \rightarrow \mathbf{h}^0_{ai}$;$[\mathbf{h}^t_{ai}, \mathbf{p}'_a, \mathbf{q}'_i] \rightarrow w^t_{ai}$ " means the hidden state is only initialized with the latent factor vector $\mathbf{u}_{ai}$ of the user-item pair in Eq. (13), and the predicted probability of the word is just calculated by Eq. (16). And "$[\mathbf{p}'_a, \mathbf{q}'_i, \phi_a, \phi_i] \rightarrow \mathbf{h}^0_{ai}$; $[\mathbf{h}^t_{ai}, \mathbf{p}'_a, \mathbf{q}'_i, \phi_a, \phi_i] \rightarrow w^t_{ai}$ " denotes the original setting of our proposed model *U-ARM*. From the experimental results, we can find the two attention modules which attentively incorporate the latent and aspect-aware user and item embeddings can help the text generation module achieve better performance.

Shared latent embeddings. In Eq. (9), Eq. (11), Eq. (14), and Eq. (15), we treat **P**=**P**' to share the latent embeddings of the users and items in both rating prediction and text generation modules. From this table, we can find that sharing the latent embedding would decrease the performance of our proposed model *U-ARM* in both tasks. This experimental results suggest that it is better not to share the latent embeddings between the rating prediction and explanation text generation modules.

5.8 Analysis of Adaptive Review Sentence Importance Learning

To study the effectiveness of the adaptive review sentence importance learning (Eq. (26)), we conduct experiments to compare the performance of *U-ARM* and *U-ARM*-E on Amazon Pet Supplies. *U-ARM*-E is the enhanced version of our proposed model with adaptive review sentence importance learning. First, we test how these two models perform on the rating prediction and explanation text generation tasks. The experimental results can refer to Table 12. Then, we conduct a case-study and user-study to check whether our proposed model *U-ARM*-E can generate more fine-grained explanation text, the experimental results can refer to Table 13 and Table 14.

Table 12. Experimental results of *U-ARM*and *U-ARM*-E on Amazon Pet Supplies. $*$ means the corresponding model achieves significant improvement with $p < 0.05$ based on the Student's t-test.

Model	RMSE	BLEU_1	BLEU_4	ROUGE_1_F	ROUGE_L_F
U-ARM	1.1032	0.2608*	0.0089*	0.3009*	0.2082*
U-ARM-E	1.1029*	0.2601	0.0077	0.2966	0.2065

Overall performance. We compare the performance of *U-ARM* , and *U-ARM*-E in Table 12. *U-ARM*-E improves the performance of the rating prediction module as the aspect extraction module performs better because it can adaptively learn better informative review for model training. And based on the BLEU and ROUGE metrics, *U-ARM*-E performs worse as the weights of the general texts are assigned to zero when calculating the text generation loss. Therefore, *U-ARM*-E prefers to generate more explanation text rather than simply generating original reviews.

Case-study. In Table 13, we randomly select three pieces of generated explanation text. From the results we can find *U-ARM*-E can generate more explanation text while *U-ARM* generates more general terms, such as general endorsements. In the first case, it is obvious that *U-ARM*-E performs better. In the second case, we can find *U-ARM*-E generates more fine-grained sentences, such as "I used it because of its very good quality" and "it's very well made". However, *U-ARM* generates only a fine-grained sentence of "it's made in China", and also many general endorsements, such as "My parroltet loves this" and "he loves it". In the third case, it is also obvious that *U-ARM*-E generates more fine-grained sentences, such as "Hide-a-squirrel is a great toy", "it is a good quality dog toy" and "it's great for their teeth", while *U-ARM* only generates one fine-grained sentence such as "Hide-a-squirrel is a great toy for my dogs". In this paper, we focus on how to generate more fine-grained explanation text from reviews. For a fair comparison with other baselines, we directly use the generated text as explanation text. When applying our proposed model in practice, we need to transform the generated text to second-persona. A simple idea to achieve this to utilize the regex expression.

User-study. In the human evaluation stage, we ask three volunteers $(1, 2, 3)$ to evaluate the performance of the models under three metrics "Fluency", "Coherence", and "Persuasiveness". "Fluency" is a sentence level metric, which reflects whether the generated sentences are fluent or not. "Coherence" is a sentence pair level metric, which reflects whether the generated sentences are coherent with the useful information from real reviews. "Persuasiveness" is a sentence context level

Table 13. Samples of the explanation text generated by *U-ARM* and *U-ARM*-E on Amazon Pet Supplies dataset. Bold fonts are used for emphasis.

	Amazon Pet Supplies
Real Review 1	This is one of the **best catnips** I have ever seen. The **scent is so nice** (even for us humans) and the cats are going crazy over this.
U-ARM	This was a **great deal** for the cats. However, they weren't interested in it.
U-ARM-E	This catnip bananas is the **best catnip** on the market.
Real Review 2	My parakeets love this bell. They love to nip at the edges of it, **fascinated** with the **little clinking sound** it makes.
U-ARM	My parrotlet loves this and it's **made in China**. I have a very small bird and he loves it.
U-ARM-E	My cocker spaniel enjoys this thing. I used it because of its **very good quality**, and it's **very well made**.
Real Review 3	Hide-a-squirrel interactive toy has been such a **big hit** with my French bulldogs that I have now purchased a hide a pet toy for each of my puppy buyers.
U-ARM	Hide-a-squirrel is a **great toy** for my dogs. I have two dogs and they love it very much.
U-ARM-E	Hide-a-squirrel is a **great toy**, and it is a **good quality** dog toy. My dogs love it, and it's **great for their teeth**.

Table 14. User study evaluation of the explanation text. "Fluency" means whether the generated sentences are fluent or not. "Coherence" means whether the generated sentences are coherent with the useful information from real reviews. "Persuasiveness" means whether the generated sentences contain useful fine-grained information to help users make decisions.

Metrics	Att2Seq	ExpansionNet	MRG	CF_GCN	*U-ARM*-E
Fluency	0.14	0.10	0.12	0.01	0.62
Coherence	0.06	0.16	0.24	0.00	0.54
Persuasiveness	0.13	0.09	0.16	0.01	0.60

metric, which reflects whether the generated sentences contain useful fine-grained information to help users make decisions. To evaluate the performance of our proposed model with human evaluation, we randomly sample 100 user-item cases. For each volunteer $A \in (1, 2, 3)$, we ask her to select the best model under each metric p for each user-item case. We use M_p^A to denote how many times the model M is selected by volunteer A under the metric p. The model M belongs to $(Att2Seq, ExpansionNet, MRG, CF\_GCN, U\text{-}ARM\text{-}E)$, and metric p belongs to ("Fluency", "Coherence", "Persuasiveness"). Then, we can calculate the performance score s_p^M of each model M based on each metric p with:

$$s_p^M = \frac{1}{3} \sum_{A \in (1,2,3)} \frac{M_p^A}{100}.$$

The evaluation result can refer to Table 14. The higher value means that the corresponding model is selected more times. From the results, we can find the generated explainable text of our proposed model *U-ARM*-E can achieve the highest score under all metrics by human evaluation.

6 CONCLUSION

In this paper, we argued that user generated reviews contain sufficient aspect semantic information to explain their ratings to items, which is useful for rating prediction and fine-grained explanation text generation. To this end, we proposed a novel *Unsupervised Aspect-aware Explainable Review based Recommendation Model (*U-ARM*)* to fully leverage aspect information for improving the quality of review based recommendation. Our key technical contributions lied in injecting the aspect information for representation learning, rating prediction, and review sentence importance learning in a unified framework for better explanation text generation. Extensive experiments on three real-world datasets demonstrated the superiority of our proposed model *U-ARM*.

7 FUTURE WORK

We believe that our proposed method can promote the development of explanation text generation tasks in recommender systems. To further boost the performance of our proposed model, we have the following plans. First, we plan to combine the advanced technologies in natural language processing for better review utilization and more fine-grained explanation text generation. Second, we only use a small proportion of historical reviews of the users and items for training, which may lead to sub-optimal results. We will study how to select valuable and high-quality reviews for better aspect-aware representations learning and achieving better performance. Third, the reviews contain both the aspects information and the customers' sentiments to different aspects of the items. It is better if we can utilize the aspect and sentiment information together.

ACKNOWLEDGEMENTS

This work was supported in part by grants from the National Natural Science Foundation of China (Grant No. 61972125, U1936219), the Foundation of Key Laboratory of Cognitive Intelligence, iFLYTEK, P.R., China(Grant No. COGOS-20190002), the Fundamental Research Funds for the Central Universities , HFUT (Grant No. JZ2020HGPA0114), and the Young Elite Scientists Sponsorship Program by CAST and ISZS.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *ICLR*.
[2] David M Blei. 2003. Latent Dirichlet Allocation. *JMLR* (2003), 993–1022.
[3] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural Attentional Rating Regression with Review-Level Explanations. In *WWW*. 1583–1592.
[4] Hanxiong Chen, Xu Chen, Shaoyun Shi, and Yongfeng Zhang. 2021. Generate Natural Language Explanations for Recommendation. In *EARS*.
[5] Lei Chen, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. 2020. Revisiting Graph Based Collaborative Filtering: A Linear Residual Graph Convolutional Network Approach. *AAAI* 34, 01 (2020), 27–34.
[6] Xu Chen, Hanxiong Chen, Hongteng Xu, Yongfeng Zhang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2019. Personalized Fashion Recommendation with Visual Explanations Based on Multimodal Attention Network: Towards Visually Explainable Recommendation. In *SIGIR*. 765–774.
[7] Yifan Chen, Yang Wang, Xiang Zhao, Jie Zou, and Maarten De Rijke. 2020. Block-Aware Item Similarity Models for Top-N Recommendation. *TOIS* 38, 4 (2020), 1–26.
[8] Zhongxia Chen, Xiting Wang, Xing Xie, Tong Wu, Guoqing Bu, Yining Wang, and Enhong Chen. 2019. Co-Attentive Multi-Task Learning for Explainable Recommendation. In *IJCAI*. 2137–2143.
[9] Zhiyong Cheng, Xiaojun Chang, Lei Zhu, Rose C. Kanjirathinkal, and Mohan Kankanhalli. 2019. MMALFM: Explainable Recommendation by Leveraging Reviews and Images. *TOIS* 37, 2 (2019), 1–28.
[10] Zhiyong Cheng, Ying Ding, Xiangnan He, Lei Zhu, Xuemeng Song, and Mohan Kankanhalli. 2018. Aˆ3NCF: An Adaptive Aspect Attention Model for Rating Prediction. In *IJCAI*. 3748–3754.
[11] Zhiyong Cheng, Ying Ding, Lei Zhu, and Mohan Kankanhalli. 2018. Aspect-Aware Latent Factor Model: Rating Prediction with Ratings and Reviews. In *WWW*. 639–648.

[12] Jin Yao Chin, Kaiqi Zhao, Shafiq Joty, and Gao Cong. 2018. ANR: Aspect-Based Neural Recommender. In *CIKM*. 147–156.
[13] Felipe Costa, Sixun Ouyang, Peter Dolog, and Aonghus Lawlor. 2018. Automatic Generation of Natural Language Explanations. In *IUI*. 1–2.
[14] Qiming Diao, Minghui Qiu, and Chao-Yuan Wu. 2014. Jointly Modeling Aspects, Ratings and Sentiments for Movie Recommendation (JMARS). In *KDD*. 193–202.
[15] Jingtao Ding, Guanghui Yu, Yong Li, Xiangnan He, and Depeng Jin. 2020. Improving Implicit Recommender Systems with Auxiliary Data. *TOIS* 38, 1 (2020), 1–27.
[16] Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. 2017. Learning to Generate Product Reviews from Attributes. In *ACL*. 623–632.
[17] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *ACL*. 1631–1640.
[18] Xinyu Guan, Zhiyong Cheng, Xiangnan He, Yongfeng Zhang, Zhibo Zhu, Qinke Peng, and Tat-Seng Chua. 2019. Attentive Aspect Modeling for Review-Aware Recommendation. *TOIS* 37, 3 (2019), 1–27.
[19] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An Unsupervised Neural Attention Model for Aspect Extraction. In *ACL*. 388–397.
[20] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW*. 173–182.
[21] Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. 2016. Convolutional Matrix Factorization for Document Context-Aware Recommendation. In *RecSys*. 233–240.
[22] 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 *SIGIR*. 275–284.
[23] Lei Li, Yongfeng Zhang, and Li Chen. 2020. Generate Neural Template Explanations for Recommendation. In *CIKM*. 755–764.
[24] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural Rating Regression with Abstractive Tips Generation for Recommendation. In *SIGIR*. 345–354.
[25] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *ACL*. 74–81.
[26] Yujie Lin, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten de Rijke. 2020. Explainable Outfit Recommendation with Joint Outfit Matching and Comment Generation. *TKDE* 32, 8 (2020), 1502–1516.
[27] Guang Ling, Michael R. Lyu, and Irwin King. 2014. Ratings Meet Reviews, a Combined Approach to Recommend. In *RecSys*. 105–112.
[28] Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies* 5, 1 (2012), 1–167.
[29] 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 *KDD*. 344–352.
[30] Yichao Lu, Ruihai Dong, and Barry Smyth. 2018. Coevolutionary Recommendation Model: Mutual Learning between Ratings and Reviews. In *WWW*. 773–782.
[31] Yichao Lu, Ruihai Dong, and Barry Smyth. 2018. Why I like It: Multi-Task Learning for Recommendation and Explanation. In *RecSys*. 4–12.
[32] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-Based Neural Machine Translation. In *ACL*. 1412–1421.
[33] Julian McAuley and Jure Leskovec. 2013. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. In *RecSys*. 165–172.
[34] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying Recommendations Using Distantly-Labeled Reviews and Fine-Grained Aspects. In *EMNLP*. 188–197.
[35] Jianmo Ni, Zachary C Lipton, Sharad Vikram, and Julian McAuley. 2017. Estimating Reactions and Recommending Products with Generative Models of Reviews. In *IJCNLP*. 783–791.
[36] Jianmo Ni and Julian McAuley. 2018. Personalized Review Generation By Expanding Phrases and Attending on Aspect-Aware Representations. In *ACL*. 706–711.
[37] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*. 311–318.
[38] Ruslan Salakhutdinov and Andriy Mnih. 2008. Probabilistic Matrix Factorization. In *NIPS*. 1–8.
[39] 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 *RecSys*. 297–305.
[40] Stefan Siersdorfer, Sergiu Chelaru, Wolfgang Nejdl, and Jose San Pedro. 2010. How Useful Are Your Comments?: Analyzing and Predicting Youtube Comments and Comment Ratings. In *WWW*. 891.

[41] Peijie Sun, Le Wu, Kun Zhang, Yanjie Fu, Richang Hong, and Meng Wang. 2020. Dual Learning for Explainable Recommendation: Towards Unifying User Preference Prediction and Review Generation. In *WWW*. 837–842.
[42] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *NIPS*. 1–9.
[43] Yunzhi Tan, Min Zhang, Yiqun Liu, and Shaoping Ma. 2016. Rating-Boosted Latent Topics: Understanding Users and Items with Ratings and Reviews. In *IJCAI*. 2640–2646.
[44] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Multi-Pointer Co-Attention Networks for Recommendation. In *KDD*. 2309–2318.
[45] Quoc-Tuan Truong and Hady Lauw. 2019. Multimodal Review Generation for Recommender Systems. In *WWW*. 1864–1874.
[46] Chong Wang and David M. Blei. 2011. Collaborative Topic Modeling for Recommending Scientific Articles. In *KDD*. 448–456.
[47] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative Deep Learning for Recommender Systems. In *KDD*. 1235–1244.
[48] Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. 2018. Explainable Recommendation via Multi-Task Learning in Opinionated Text Data. In *SIGIR*. 165–174.
[49] Libing Wu, Cong Quan, Chenliang Li, Qian Wang, Bolong Zheng, and Xiangyang Luo. 2019. A Context-Aware User-Item Representation Learning for Item Recommendation. *TOIS* 37, 2 (2019), 1–29.
[50] Yao Wu and Martin Ester. 2015. FLAME: A Probabilistic Model Combining Aspect Based Opinion Mining and Collaborative Filtering. In *WSDM*. 199–208.
[51] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*. 2048–2057.
[52] Bao Yang, Fang Hui, and Zhang Jie. 2014. TopicMF Simultaneously Exploiting Ratings and Reviews for Recommendation. In *AAAI*. 2–8.
[53] Wei Zhang and Jianyong Wang. 2016. Integrating Topic and Latent Factors for Scalable Personalized Review-Based Rating Prediction. *TKDE* 28, 11 (2016), 3013–3027.
[54] Wei Zhang, Quan Yuan, Jiawei Han, and Jianyong Wang. 2016. Collaborative Multi-Level Embedding Learning from Reviews for Rating Prediction. In *IJCAI*. 2986–2992.
[55] Yongfeng. Zhang and Xu. Chen. 2020. Explainable Recommendation: A Survey and New Perspectives. *Now Foundations and Trends* (2020).
[56] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit Factor Models for Explainable Recommendation Based on Phrase-Level Sentiment Analysis. In *SIGIR*. 83–92.
[57] Lei Zheng, Vahid Noroozi, and Philip S. Yu. 2017. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In *WSDM*. 425–434.