Aspect-guided Syntax Graph Learning for Explainable Recommendation

Yidan Hu, Yong Liu, Chunyan Miao, Gongqi Lin, Yuan Miao

Abstract—Explainable recommendation systems provide explanations for recommendation results to improve their transparency and persuasiveness. The existing explainable recommendation methods generate textual explanations without explicitly considering the user's preferences on different aspects of the item. In this paper, we propose a novel explanation generation framework, namely **A**spect-guided **E**xplanation generation with **S**yntax **G**raph (**AESG**), for explainable recommendation. Specifically, AESG employs a review-based syntax graph to provide a unified view of the user/item details. An aspect-guided graph pooling operator is proposed to extract the aspect-relevant information from the review-based syntax graphs to model the user's preferences on an item at the aspect level. Then, an aspect-guided explanation decoder is developed to generate aspects and aspect-relevant explanations based on the attention mechanism. The experimental results on three real datasets indicate that AESG outperforms state-of-the-art explanation generation methods in both single-aspect and multi-aspect explanation generation tasks, and also achieves comparable or even better preference prediction accuracy than strong baseline methods.

Index Terms—Explanation generation, explainable recommendation, hierarchical graph pooling.

1 INTRODUCTION

RECOMMENDATION systems have been widely used to help users make decisions by suggesting them a list of items they may have interests. Various types of recommendation methods have been developed based on collaborative filtering [1] and deep learning techniques [2]. Although these methods can usually achieve satisfactory performances, it is still very hard to explain their recommendation mechanisms. Thus, many recent research efforts [3], [4], [5], [6], [7] have been devoted to building explainable recommendation models that explain why an item is recommended by generating high-quality explanations, which can help improve the transparency and persuasiveness of recommendation systems. In practice, different strategies may be adopted to explain the recommendation results, *e.g.*, images, the behaviors of relevant users, and textual descriptions of relevant items [8]. In this work, we focus on generating high-quality review text to explain the recommendation results presented to the user.

The review generation methods for explainable recommendation can be roughly classified into two groups: template-based approaches and natural language generation approaches [8]. The template-based methods generate explanations by filling the generated words in a sentence template. For example, in the template *"You might be interested in [**aspect**], on which this product performs well"*, we can replace *[**aspect**]* by a generated aspect to produce text explanation for item recommendation [9]. However, template-based explanations are uninformative and not persuasive. Moreover, designing high-quality templates usually requires domain knowledge. Natural language generation approaches can generate more natural and flexible sentences. Such approaches have recently attracted increasing research attention. However, the pioneer works [10], [11] focus on generating short reviews or tips only based on given attributes (*e.g.*, user ID, item ID, and rating value). Thus, it is difficult for them to generate reliable explanations without considering other generative signals [9], [11].

Aspects, an important type of generative signal, which usually represent item features (*e.g.*, "price" and "romance"), have recently been exploited to build aspect-aware explanation generation models [12], [13], [14]. In these methods, the aspects are extracted from the user-generated reviews and used to train the explanation generation model. These methods assume the user's interested aspects are available for explanation generation. In practice, this assumption usually does not hold, because we need to predict the user's preferences for different aspects of a target item in many application scenarios. The aspect-aware generation framework [15] provides a potential solution to address this problem. However, this method only considers the user ID, item ID, and the rating value as inputs. Thus, it cannot be effective in capturing the aspect-relevant details of the user and item for generating long and informative explanations.

To address this problem, we build a review-based syntax graph to provide a unified view of the user/item details based on the review data. Firstly, a syntax dependency tree is built from each review. The relations in the dependency tree provide important clues to mine aspects, details, and opinions. From Figure 1 (a), we observe two sub-trees: *story* $\xrightarrow{\text{nmod:with}}$ *twists* $\xrightarrow{\text{amod}}$ *interesting*, *story* $\xrightarrow{\text{nmod:with}}$ *characters* $\xrightarrow{\text{amod}}$ *memorable*. They are both built up with the structure

- *Yidan Hu and Chunyan Miao are currently with School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798. Email: yidan001@e.ntu.edu.sg, ascymiao@ntu.edu.sg.*
- *Yong Liu is with Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), Nanyang Technological University, Singapore 639798. Email: stephenliu@ntu.edu.sg.*
- *Yuan Miao and Gongqi Lin are currently with IT Discipline, Victoria University, Australia. Email: yuan.miao@vu.edu.au, lingongqi2009@gmail.com*

Manuscript received XXX; revised XXX.

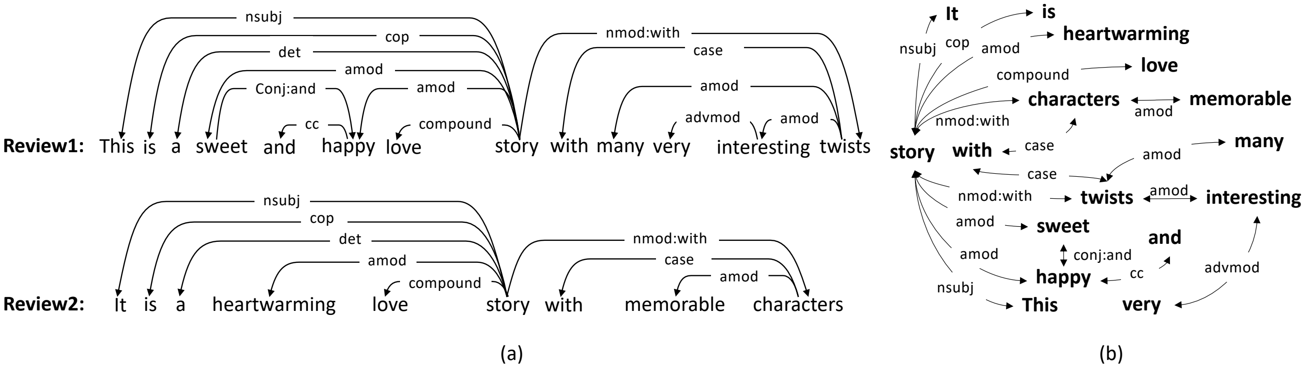


Fig. 1: An example of the review-based syntax graph. (a) shows the dependency tree structures of two reviews. (b) shows the review-based syntax graph built based on the nodes and relations extracted from the dependency trees of reviews.

of *aspect → details → opinion*. To aggregate the details of the same aspect in different reviews, we construct the review-based syntax graph by connecting details in different dependency trees. As shown in Figure 1 (b), details about *love story*, such as *characters* and *twists*, can be directly connected to *love story* in the review-based syntax graph.

Moreover, we propose a novel explainable recommendation framework, *i.e.,* Aspect-guided Explanation generation with Syntax Graph (AESG). Specifically, AESG performs hierarchical aspect-guided graph pooling on the user/item review-based syntax graph to extract the aspect-relevant information for building the user/item representation. The user's interests are matched with the item properties at the aspect level to predict her preferences for different aspects of the item. Then, an aspect-guided explanation decoder is developed to generate aspects and aspect-relevant explanations based on attention mechanisms. To demonstrate the effectiveness of the proposed AESG model, we perform extensive experiments on three real-world datasets. The experimental results indicate that AESG outperforms state-of-the-art explanation generation methods, and achieves comparable or even better preference prediction accuracy than baseline methods.

2 Related Work

This section reviews the most relevant works about explainable recommendation methods, review-based recommendation methods, and graph-based recommendation methods.

2.1 Explainable Recommendation Methods

Explainable recommendation has recently attracted a lot of research attentions. Various methods have been proposed to provide different types of explanations for the recommendation results [8], such as feature-based explanations [16], textual sentence explanations [14], [17], [18], visual explanations [19], [20], [21], and social explanations [22], [23]. This work mainly focuses on the explainable recommendation methods that generate textual sentence explanations for the recommendation results.

In the literature, there are two main groups of text-based explanation generation methods for explainable recommendation, namely template-based and generation-based methods. Template-based methods generate recommendation explanations by filling pre-defined templates with different words for different users. For example, the explicit factor model [9] generates the explanations by filling in the aspect in the pre-defined template. Moreover, [17] introduces a template-based explainable recommendation model with aspects and opinions. However, these methods cannot provide more details about user preferences, and manually designing the template is also time-consuming.

Generation-based methods focus on developing natural language generation methods for explainable recommendation. For example, [10], [11], [24] employ Recurrent Neural Networks (RNNs) based methods to encode the attribute information, *e.g.,* user ID, item ID, and rating value, to generate explanations for recommendation results. Moreover, [25] uses generative adversarial networks to provide personalized explanations for each user. However, these methods cannot generate reliable and precise explanations due to the lack of guided information. To address this problem, other auxiliary information has been exploited to generate explanations from given aspects [12]. For example, [13] introduces the reference-based seq2seq model that treats historical justifications as references. [18] utilizes a Transformer-based generation method to exploit both IDs and item aspects for explainable recommendation. [26] employs an unsupervised neural aspect extraction model to learn the aspect representation and exploit aspect information in the explanation generation process. In addition, the neural template-based explanation generation framework [14] has also been developed to integrate the advantages of both the template-based and language generation methods. It first uses the given aspect as a template and then generates template-controlled explanations.

These aforementioned methods usually consider the user's interested aspects as extra information to generate controlled explanations. However, it is challenging to detect a user's preference on different aspects. To solve this problem, [15] proposes a coarse-to-fine generation framework, which first generates a sentence skeleton and then generates the aspect-aware explanations. In addition, [27] presents a knowledge enhanced review generation model that exploits knowledge graph (KG) information to generate aspect-aware explanations. One main limitation of these methods is that they can not explicitly generate specific aspects. Differing from existing methods, the proposed AESG model exploits review-based syntax graphs from review data to first predict users' preference on different aspects, and then

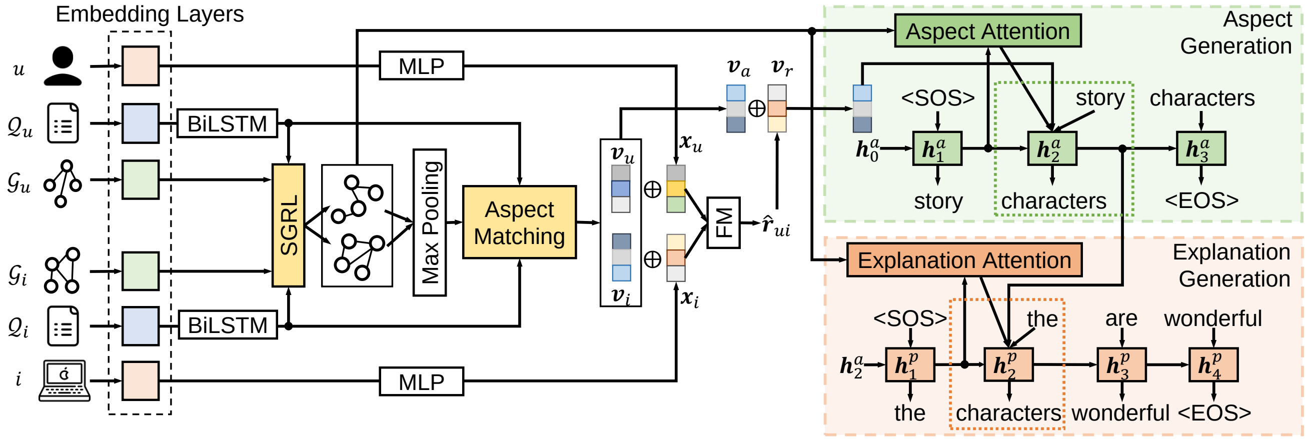


Fig. 2: Overall structure of the proposed AESG model. In this figure, we use the aspect and explanation generation processes of a single word ”characters” as an example.

generate aspects and aspect-relevant explanations.

2.2 Review-based Recommendation Methods

The user-generated review data have been widely studied to enhance the recommendation performance [4], [28], [29]. For example, [28] employs two parallel Convolutional Neural Networks (CNN) and a shared layer to exploit the review data to improve the learned user and item representations. The recent studies [30], [31], [32] apply attention mechanisms to improve recommendation performance by searching informative text from users’ textual reviews. For example, in [33], a dual attention-based CNN is used to combine both local attention and global attention to find the key information from reviews. In [4], the attention mechanism is used to select useful review text from existing review data by weighing their importance, and this extra information is then incorporated to predict rating scores. Moreover, [29] designs a local attention layer and a mutual attention layer to jointly learn the features from the user reviews and model the user-item interactions.

2.3 Graph-based Recommendation Methods

Graph Neural Networks (GNN) are effective in exploiting graph structures to improve recommendation performance [34]. For example, [35] uses efficient random walk and graph convolution algorithms to learn item representations, considering both graph structure and item features. [36] employs graph convolutional networks to extract high-order relations from the user-item bipartite interaction graph. Moreover, some recent works [32], [37], [38] build graphs from review text and use GNN to extract the semantic information from reviews. For instance, [39] introduces a graph-based contrastive learning framework that exploits review information to enhance the user-item interaction graph for improving recommendation performance. Moreover, [40] also shows that GNN can help improve the performance of sequential recommendation models. For example, [41] uses GNN to model users’ session-based behavior sequences. [42] applies GNN to exploit context information from the global item transition graph and each session graph for sequential recommendation. In this work, the proposed model designs aspect-guided graph pooling to learn the syntax information, and capture the user preference and item properties from the review-based syntax graph.

3 Preliminaries

In this section, we introduce some background about the construction of review-based syntax graph and the research problem studied in this work.

3.1 Review-based Syntax Graph Construction

In this work, we denote the historical reviews of a user u by $\mathcal{D}_u = \{d_u^1, d_u^2, \cdots, d_u^{n_{du}}\}$ and the historical reviews of an item i by $\mathcal{D}_i = \{d_i^1, d_i^2, \cdots, d_i^{n_{di}}\}$, where n_{du} and n_{di} denote the number of reviews associated with u and i, respectively. To better understand the review data, we build a user review-based syntax graph $\mathcal{G}_u$ and an item review-based syntax graph $\mathcal{G}_i$ from the review sets $\mathcal{D}_u$ and $\mathcal{D}_i$, respectively. For each user u, we first apply text pre-processing techniques (*e.g.*, tokenization and spelling checker) on $\mathcal{D}_u$. Then, dependency parsing [43] is used to automatically generate constituent-based representation (*i.e.*, dependency tree) for each review sentence based on syntax. As shown in Figure 1 (a), there are two dependency trees built from Review 1 and Review 2, respectively. A syntax relation drawn from a fixed inventory of grammatical relations set [44] connects two nodes in the dependency tree. We perform pruning to remove relations with little aspect-relevant information, instead of directly removing words. Specifically, we remove the following three relations from the dependency tree: 1) “det” relation that refers to the determiner relationship between two words, 2) “punct” relation that refers to punctuation in the sentence, 3) “nmod:poss” relation that refers to the possessives relationship between two words. Next, isolated words will be removed. Then, we leverage the same words as connection nodes to connect different dependency trees built from sentences in $\mathcal{D}_u$. After that, we can obtain the user review-based syntax graph $\mathcal{G}_u = \{\mathcal{X}_u, \mathcal{E}_u\}$, where $\mathcal{X}_u$ denotes the set of nodes (*i.e.*, words), and $\mathcal{E}_u$ denotes the set of edges $\mathcal{E}_u = \{(x_h, r, x_t)|x_h, x_t \in \mathcal{X}_u, r \in \mathcal{R}\}$. Here, r denotes the relation connecting the two nodes, and $\mathcal{R}$ is the set of all possible relations. Similar operations are performed on the item reviews $\mathcal{D}_i$ to obtain the item review-based syntax graph $\mathcal{G}_i = \{\mathcal{X}_i, \mathcal{E}_i\}$.

By connecting the same words in different dependency trees extracted from reviews of the same user/item, we can build relationships between different review sentences that have similar content. Then, each review sentence can be described by a subgraph of the syntax graph. If two sentences share more common words, there will exist more connections between their corresponding subgraphs. Thus, the review-based syntax graph provides a more complex structure that can describe the content of each review sentence, as well as the relationships between similar review sentences. Similar operations have also been applied in previous studies [45], [46] to obtain text representations.

3.2 Problem Formulation

Following [14], [47], we first extract aspects from the historical review data using the tool developed in [9]. For each user u, we extract aspects from $\mathcal{D}_u$. Then, we sort the extracted aspects according to their occurrence frequency in descending order, and choose the top-n ranked aspects to describe the user properties, which are denoted by $\mathcal{Q}_u = \{q_u^1, q_u^2, \cdots, q_u^n\}$. Similarly, for each item i, we can obtain its top-n most frequent aspects extracted from $\mathcal{D}_i$, and denote them by $\mathcal{Q}_i = \{q_i^1, q_i^2, \cdots, q_i^n\}$. In this work, we study the following explainable recommendation problem: *given a user u and an item i, their historical aspect sets $\mathcal{Q}_u$ and $\mathcal{Q}_i$, and review-based syntax graphs $\mathcal{G}_u$ and $\mathcal{G}_i$, we aim to predict the user's preference r_{ui} on the item, mine the user's interested aspects $\mathcal{A}_{ui} = \{a_1, a_2, \cdots, a_m\}$ of the item, and generate aspect-aware explanations $\mathcal{P}_{ui} = \{p_1, p_2, \cdots, p_m\}$, in the form of a set of sentences.*

4 The Proposed AESG Framework

Figure 2 shows the overall framework of AESG, which contains the following main components: 1) syntax graph representation learning (SGRL), 2) aspect matching, 3) preference prediction, and 4) aspect-guided explanation decoder. Next, we introduce the details of each component.

4.1 Syntax Graph Representation Learning

The objective of the SGRL module is to extract representations for each review-based syntax graph. Recall that the aspects in $\mathcal{Q}_u$ and $\mathcal{Q}_i$ are ranked in descending order based on their occurrence frequency. We argue that the more frequently an aspect appears in the reviews of a user/item, the more important it is to the user/item. Thus, we firstly apply a BiLSTM f to encode the word embedding of an aspect q, and obtain output backward hidden state as the representation $f(q)$. The BiLSTM can capture the position information in the ranking list, which indicates the importance of aspects. Then, hierarchical aspect-guided graph pooling is used to extract the review-based syntax graph representation at the aspect level. We propose an aspect-guided graph pooling (AGP) operator to effectively extract the aspect-specific knowledge from the review-based syntax graph. AGP employs the user/item historical aspects to guide review-based syntax graph representation learning.

4.1.1 *Aspect-guided Graph Pooling Operator*

Figure 3 (a) shows the workflow of the AGP operator. The inputs of an AGP operator include a graph $\mathcal{G} = \{\mathcal{X}, \mathcal{E}, \mathbf{X}, \mathbf{A}\}$ and an aspect q with its representation $f(q)$. Here, $\mathcal{X}$ and $\mathcal{E}$ denote the set of nodes and edges in $\mathcal{G}$, respectively. $\mathbf{X}$ is the node feature matrix and $\mathbf{A}$ is the adjacency matrix. Firstly, a graph attention network (GAT) [48] is used to encode graph $\mathcal{G}$. For each node $x_h \in \mathcal{X}$, the feature of x_h is updated by aggregating the input features of neighborhood nodes and adding its input feature $\mathbf{x}_h$ by self-loop as,

$$\overline{\mathbf{x}}_h = \mathbf{x}_h + \sum_{x_t \in \mathcal{N}_h} \alpha(x_h, r, x_t)\mathbf{x}_t \mathbf{W}_1, \tag{1}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_0 \times d_0}$ is a learnable weight matrix, $\mathcal{N}_h$ denotes the set of first-hop neighbors of x_h in the graph, and $\alpha(x_h, r, x_t)$ denotes the attention score between two nodes x_t and x_h. Following [48], we define the attention score $\alpha(x_h, r, x_t)$ as,

$$\alpha(x_h, r, x_t) = \frac{\exp\left[\pi(x_h, r, x_t)\right]}{\sum_{x_{t'} \in \mathcal{N}_h} \exp\left[\pi(x_h, r, x_{t'})\right]}. \tag{2}$$

Here, $\pi(x_h, r, x_t)$ is implemented by the following attentional mechanism,

$$\pi(x_h, r, x_t) = \sigma_1\big((\mathbf{x}_h \mathbf{W}_2)(\mathbf{x}_t \mathbf{W}_3 + \mathbf{r}\mathbf{W}_4)^\top\big), \tag{3}$$

where $\mathbf{r}$ is the embedding of the relation r, $\sigma_1(\cdot)$ is LeakyReLU activation function, $\mathbf{W}_2$, $\mathbf{W}_3$, $\mathbf{W}_4 \in \mathbb{R}^{d_0 \times d_0}$ are learnable weight matrixes. We use a matrix $\overline{\mathbf{X}}$ to denote the updated features of all nodes.

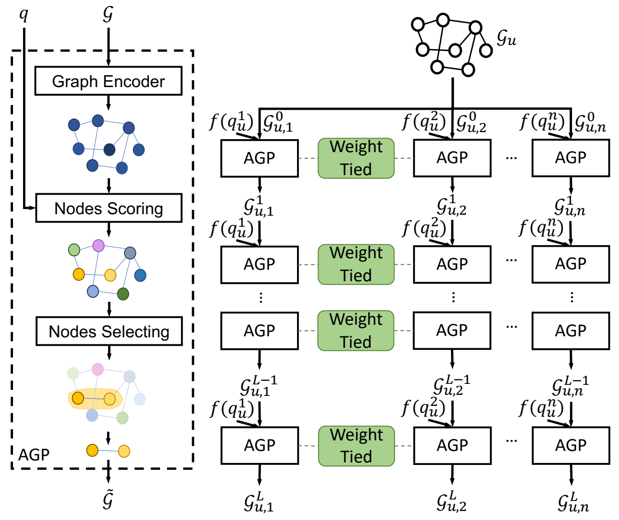
Inspired by previous graph pooling methods [49], [50], [51], we define the following aspect-aware importance score to describe the relevance of each node in $\mathcal{G}$ to the given aspect q,

$$\delta(x_h) = \mathrm{abs}\big(\overline{\mathbf{x}}_h f(q)^\top\big), \tag{4}$$

where abs$(\cdot)$ denotes the absolute value function. The nodes in $\mathcal{G}$ can be ranked according to $\delta(x_h)$ in descending order. Then, we denote the set of top-K ranked nodes by $\widetilde{\mathcal{X}}$ and their indices by $\mathcal{I}$. In this work, we empirically set $K = \lceil \rho|\mathcal{X}| \rceil$, where ρ is the pooling ratio, $|\cdot|$ denotes the cardinality of a set, $\lceil \cdot \rceil$ is the ceiling function. The new features of nodes in $\widetilde{\mathcal{X}}$ and adjacency matrix $\widetilde{\mathbf{A}}$ of the corresponding graph are defined as,

$$\widetilde{\mathbf{X}} = \mathrm{ReLU}\big(\overline{\mathbf{X}}[\mathcal{I}, :]\mathbf{W}_5 + \mathbf{b}_1\big), \quad \widetilde{\mathbf{A}} = \mathbf{A}[\mathcal{I}, \mathcal{I}], \tag{5}$$

where $\mathbf{W}_5 \in \mathbb{R}^{K \times d_0}$ and $\mathbf{b}_1 \in \mathbb{R}^{1 \times d_0}$ are the weight matrix and bias vector respectively. $\overline{\mathbf{X}}[\mathcal{I}, :]$ is row-wise indexed feature matrix. $\mathbf{A}[\mathcal{I}, \mathcal{I}]$ aims to obtain the row-wise and column-wise indexed adjacency matrix from $\mathbf{A}$. Then, $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{A}}$ are the new feature matrix and the corresponding adjacency matrix after pooling. Moreover, we use $\widetilde{\mathcal{E}}$ to denote the set of edges that describe the connecting relationships between the nodes in $\widetilde{\mathcal{X}}$. Then, the output of the AGP operator is denoted by $\widetilde{\mathcal{G}} = \{\widetilde{\mathcal{X}}, \widetilde{\mathcal{E}}, \widetilde{\mathbf{X}}, \widetilde{\mathbf{A}}\}$.



(a) (b)

Fig. 3: (a) shows the flow of the AGP operator where q and $\mathcal{G}$ denote the input aspect and graph. (b) shows the flow of review-based syntax graph representation learning.

4.1.2 Hierarchical Graph Pooling

We perform hierarchical graph pooling on the item/user review-based syntax graph, by staking AGP operators. Here, we only introduce the hierarchical graph pooling on the user review-based syntax graph $\mathcal{G}_u$. The same process is also followed on the item review-based syntax graph $\mathcal{G}_i$. As shown in Figure 3 (b), we conduct L layers graph pooling on $\mathcal{G}_u$, guided by the user aspect set $\mathcal{Q}_u = \{q_u^1, q_u^2, \cdots, q_u^n\}$.

At the ℓ-th layer, there are n AGP operators. The input graph of the k-th AGP operator is $\mathcal{G}_{u,k}^{\ell-1} = \{\mathcal{X}_{u,k}^{\ell-1}, \mathcal{E}_{u,k}^{\ell-1}, \mathbf{X}_{u,k}^{\ell-1}, \mathbf{A}_{u,k}^{\ell-1}\}$. The representation $f(q_u^k)$ of aspect q_u^k is used to guide the pooling in the k-th AGP operator. The output graph of this AGP operator is $\mathcal{G}_{u,k}^{\ell}$. Note that the inputs for the first layer $\mathcal{G}_{u,k}^0 = \mathcal{G}_u$ for $k = 1, 2, \cdots, n$. We perform max pooling on $\mathbf{X}_{u,k}^{\ell}$ to obtain the aspect-aware graph representation $\mathbf{g}_{u,k}^{\ell}$ at the ℓ-th layer. After performing the above pooling operation L times, we can obtain multiple representations of $\mathcal{G}_u$ that are relevant to the aspect q_u^k, *i.e.*, $\mathbf{g}_{u,k}^1, \mathbf{g}_{u,k}^2, \cdots, \mathbf{g}_{u,k}^L$. To fuse the graph representations from fine-grained to coarse, we concatenate these representations to form the representation $\mathbf{g}_u^k$ of the review-based syntax graph $\mathcal{G}_u$ as $\mathbf{g}_u^k = \mathbf{g}_{u,k}^1 \oplus \mathbf{g}_{u,k}^2 \oplus \cdots \mathbf{g}_{u,k}^L$. Similarly, we can define the representation $\mathbf{g}_i^k$ of the review-based syntax graph $\mathcal{G}_i$ for an item i.

4.2 Aspect Matching

To better describe the user's preference on different aspects, we compute aspect-level user representation $\mathbf{S}_u$ and item representation $\mathbf{S}_i$ by concatenating the representations of historical aspects with those of the review-based syntax graphs as follows,

$$\mathbf{S}_u = \begin{bmatrix} \mathbf{g}_u^1 \oplus f(q_u^1) \\ \mathbf{g}_u^2 \oplus f(q_u^2) \\ \cdots \\ \mathbf{g}_u^n \oplus f(q_u^n) \end{bmatrix}, \mathbf{S}_i = \begin{bmatrix} \mathbf{g}_i^1 \oplus f(q_i^1) \\ \mathbf{g}_i^2 \oplus f(q_i^2) \\ \cdots \\ \mathbf{g}_i^n \oplus f(q_i^n) \end{bmatrix}. \quad (6)$$

Following [52], [53], we use $\mathbf{S}_u$ and $\mathbf{S}_i$ as features to define the following aspect level importance weight matrix,

$$\mathbf{M}_{u,i} = \mathrm{ReLU}(\mathbf{S}_u \mathbf{W}_s \mathbf{S}_i^\top), \quad (7)$$

where $\mathbf{W}_s \in \mathbb{R}^{d' \times d'}$ is a learnable weight matrix, and $d' = d_0 \times L + d_1$. In $\mathbf{M}_{u,i} \in \mathbb{R}^{n \times n}$, each element $\mathbf{M}_{u,i}[x, y]$ describes the importance of the y-th item aspect to the x-th user aspect. Then, we fuse the aspect level information of the user and item with $\mathbf{M}_{u,i}$ as follows,

$$\mathbf{v}_u = \phi\Big(\mathbf{S}_u \mathbf{W}_u^s \oplus \mathbf{M}_{u,i}(\mathbf{S}_i \mathbf{W}_i^s)\Big), \quad (8)$$

$$\mathbf{v}_i = \phi\Big(\mathbf{S}_i \mathbf{W}_i^s \oplus \mathbf{M}_{u,i}^\top(\mathbf{S}_u \mathbf{W}_u^s)\Big), \quad (9)$$

where $\mathbf{W}_u^s$, $\mathbf{W}_i^s \in \mathbb{R}^{n \times d'}$ are trainable parameters, and $\phi(\cdot)$ denotes the mean pooling operation. Here, $\mathbf{v}_u$ aims to incorporate user interested aspects and the user preferences for the item aspect. Similarly, $\mathbf{v}_i$ aims to combine item typical aspect and the representation of aspects that are highly relevant to the user.

4.3 Preference Prediction

For each user u, we feed the user ID embedding $\mathbf{e}_u$ into a Multi-Layer Perceptron (MLP) and concatenate its output with her aspect level preference vector $\mathbf{v}_u$ to form the final representation as follows,

$$\mathbf{x}_u = \mathrm{MLP}_u(\mathbf{e}_u) \oplus \mathbf{W}_u^v \mathbf{v}_u, \quad (10)$$

where $\mathbf{W}_u^v \in \mathbb{R}^{(2d') \times d_2}$ is a learnable parameter. Similarly, we can obtain the final representation $\mathbf{x}_i$ of the item i. Then, we concatenate $\mathbf{x}_u$ and $\mathbf{x}_i$ to form $\mathbf{x}$. A Factorization Machine (FM) layer [54] is applied to predict the user u's preference on the item i as follows,

$$\hat{r}_{ui} = b_0 + b_u + b_i + \mathbf{x}\mathbf{w}^\top + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j, \quad (11)$$

where b_0, b_u and b_i are global bias, user bias, and item bias. $\mathbf{v}_i$ and $\mathbf{v}_j$ are the i-th and j-th variants. $\mathbf{w} \in \mathbb{R}^{1 \times (4d_2)}$ is coefficient vector, and $\langle \cdot, \cdot \rangle$ is the dot product of two vectors.

4.4 Aspect-guided Explanation Decoder

In AESG, we adopt two attention-based Long Short-Term Memory (LSTM) models as the aspect decoder and explanation decoder, respectively. This section introduces the details of these two decoders.

4.4.1 Aspect Decoder

As shown in Figure 3, for each aspect q_u^k, we can obtain the node feature matrix $\mathbf{X}_{u,k}^L$ of the graph $\mathcal{G}_{u,k}^L$ after graph pooling on the user review-based syntax graph. Similarly, we can also obtain the node feature matrix $\mathbf{X}_{i,k}^L$ after graph pooling on the item review-based syntax graph. Then, we stack all aspect-relevant node feature matrices to form the following matrices,

$$\mathbf{X}_u = \begin{bmatrix} \mathbf{X}_{u,1}^L \\ \cdots \\ \mathbf{X}_{u,n}^L \end{bmatrix}, \quad \mathbf{X}_i = \begin{bmatrix} \mathbf{X}_{i,1}^L \\ \cdots \\ \mathbf{X}_{i,n}^L \end{bmatrix}. \quad (12)$$

Note that, in $\mathbf{X}_u$ and $\mathbf{X}_i$, each row denotes a node feature vector. To initial the hidden state, we first map the predicted rating $\hat{r}_{ui}$ into a sentiment representation $\mathbf{v}_r$ to guide aspect and explanation generation as,

$$\mathbf{v}_r = \text{ReLU}(\mathbf{W}_u^r \hat{r}_{ui} + \mathbf{b}_u^r), \tag{13}$$

where $\mathbf{W}_u^r \in \mathbb{R}^{1\times d_1}$ and $\mathbf{b}_u^r \in \mathbb{R}^{1\times d_1}$ are a trainable weight matrix and a bias vector. Then, we feed $\mathbf{v}_r$, $\mathbf{v}_u$, and $\mathbf{v}_i$ into an MLP layer as,

$$\mathbf{h}_0^a = \text{MLP}_h(\mathbf{v}_r \oplus \mathbf{v}_u \oplus \mathbf{v}_i). \tag{14}$$

To fully exploit the review-based syntax graph information, at each decoding time-step $j-1$, we incorporate hidden state $\mathbf{h}_{j-1}^a$ and each node feature $\mathbf{x}_z \in \mathbf{X}_u$ (*i.e.*, each row in $\mathbf{X}_u$) to calculate attention vector $\mathbf{c}_{u,j-1}^a$ as,

$$\begin{aligned}\beta_{u,z}^{j-1} &= \frac{\exp(\text{MLP}(\mathbf{x}_z \oplus \mathbf{h}_{j-1}^a))}{\sum_{\mathbf{x}_{\acute{z}}\in\mathbf{X}_u}\exp(\text{MLP}(\mathbf{x}_{\acute{z}} \oplus \mathbf{h}_{j-1}^a))},\\ \mathbf{c}_{u,j-1}^a &= \sum_{\mathbf{x}_z\in\mathbf{X}_u}\beta_{u,z}^{j-1}\mathbf{x}_z,\end{aligned} \tag{15}$$

where $\beta_{u,z}^{j-1}$ is a scalar describing the correlation between the node feature $\mathbf{x}_z$ and the hidden state $\mathbf{h}_{j-1}^a$, and $\mathbf{c}_{u,j-1}^a$ is a weighted sum of all the node features in $\mathbf{X}_u$.

Similarly, we can obtain the attention vector $\mathbf{c}_{i,j-1}^a$ from $\mathbf{X}_i$. The next time-step hidden state $\mathbf{h}_j^a$ is,

$$\begin{aligned}\hat{\mathbf{h}}_{j-1}^a &= \mathbf{h}_{j-1}^a \oplus \mathbf{c}_{u,j-1}^a \oplus \mathbf{c}_{i,j-1}^a \oplus \mathbf{h}_0^a,\\ \mathbf{h}_j^a &= \text{LSTM}(\hat{\mathbf{h}}_{j-1}^a, E(w_{j-1}^a)),\end{aligned} \tag{16}$$

where the $E(w_{j-1}^a)$ is the embedding of the previous aspect w_{j-1}^a. $\mathbf{h}_j^a$ is fed into an MLP layer to obtain the probability of target aspect w_j^a as,

$$\mathcal{P}(w_j^a) = \text{softmax}(\mathbf{W}_a\mathbf{h}_j^a + \mathbf{b}_a), \tag{17}$$

where $\mathbf{W}_a \in \mathbb{R}^{d_1\times d_a}$ is a trainable parameter, d_a is the size of the aspect vocabulary.

4.4.2 *Explanation Decoder*

After the user interested aspect set w^a has been generated, we can further generate aspect-relevant explanations. For the j-th explanation $p_j \in P$, we utilize the j-th hidden state $\mathbf{h}_j^a$ of aspect encoder as the initial hidden state $\mathbf{h}_{j,0}^p$. Following Eq. (15) and Eq. (16), we further use the review-based syntax graphs obtain attention vector $\mathbf{c}_{u,j,t-1}^p$ and $\mathbf{c}_{i,j,t-1}^p$ at each decoding time-step $t-1$. We also concatenate $\mathbf{h}_{j,0}^p$, $\mathbf{c}_{u,j,t-1}^p$, $\mathbf{c}_{i,j,t-1}^p$, and previous hidden state to obtain $\hat{\mathbf{h}}_{j,t-1}^p$. Then, $\hat{\mathbf{h}}_{j,t-1}^p$ and the embedding of previous predicted word $E(w_{j,t-1}^p)$ are fed into the decoder as,

$$\mathbf{h}_{j,t}^p = \text{LSTM}(\hat{\mathbf{h}}_{j,t-1}^p, E(w_{j,t-1}^p)). \tag{18}$$

The probability of target word $w_{j,t}^p$ is calculated as,

$$\mathcal{P}(w_{j,t}^p) = \text{softmax}(\mathbf{W}_p\mathbf{h}_{j,t}^p + b_p), \tag{19}$$

where $\mathbf{W}_p \in \mathbb{R}^{d_1\times d_v}$ is a weight parameter and d_v is the size of the vocabulary.

4.5 Multi-task Learning Objective Function

For the explanation generation task, we define the following cross-entropy losses for aspect and explanation generation respectively,

$$\begin{aligned}\ell_{ui}^{(a)} &= \frac{1}{|\mathcal{A}_{ui}|}\sum_{j=1}^{|\mathcal{A}_{ui}|} -\log(\mathcal{P}(w_j^a)),\\ \ell_{ui}^{(p)} &= \frac{1}{|\mathcal{P}_{ui}|}\sum_{j=1}^{|\mathcal{P}_{ui}|}\frac{1}{|p_j|}\sum_{t=1}^{|p_j|} -\log(\mathcal{P}(w_{j,t}^p)),\end{aligned} \tag{20}$$

where $|\mathcal{A}_{ui}|$ and $|\mathcal{P}_{ui}|$ denote the length of ground-truth aspect and explanation sets for a given user-item pair (u,i). $|p_j|$ denotes the length of the ground-truth explanation for the j-th aspect. $\mathcal{P}(w_j^a)$ and $\mathcal{P}(w_{j,t}^p)$ denote the probability of aspect w_j^a and word $w_{j,t}^p$. In addition, we also choose preference prediction as an auxiliary task to learn the AESG model and define the loss function as follows,

$$\ell_{ui}^{(r)} = (\hat{r}_{ui} - r_{ui})^2, \tag{21}$$

where $\hat{r}_{ui}$ and r_{ui} denote the predicted and ground-truth rating values respectively. The final loss function of the proposed AESG model is defined as follows,

$$\frac{1}{|\mathcal{O}|}\sum_{(u,i)\in\mathcal{O}}\ell_{ui}^{(a)} + \ell_{ui}^{(p)} + \ell_{ui}^{(r)}, \tag{22}$$

where $\mathcal{O}$ denotes the set of observed user-item pairs in the training data, and $|\mathcal{O}|$ is the cardinality of set $\mathcal{O}$. The entire framework can be effectively trained by minimizing Equation (22) using end-to-end back propagation.

5 Experiments

In this work, we perform experiments to evaluate both the explanation generation performance and preference prediction performance of the proposed AESG model.

5.1 Experimental Settings

5.1.1 *Experimental Datasets*

The experiments are conducted on the Amazon Review dataset [55] and Yelp Challenge 2019 dataset[1], which have been widely used for explanation generation. For the Amazon review dataset, we choose the following 5-core subsets for evaluation: ”Kindle Store” and ”Electronics” (respectively denoted by Kindle and Elec.). For the Yelp dataset and Amazon review dataset, we keep users and items that have more than 20 and 5 reviews for experiments respectively, due to the limitation of computation resources. In each dataset, a record consists of user ID, item ID, overall rating, and textual review. Following previous studies [47], [56], we first extract aspects from the review data by the tool developed in [9]. Then, we only keep records that contain more than one aspect and extract aspect-relevant sentences from reviews as target explanations. Table 1 summarizes the statistics of experimental datasets. For each dataset, we randomly split the data into training, validation, and test data by the ratio 8:1:1.

[1] https://www.yelp.com/dataset/challenge

Dataset	# Users	# Items	# Reviews	# Aspects
Kindle	44,589	46,691	650,575	2,683
Elec.	118,607	45,334	1,036,965	7,345
Yelp	22,982	15,525	1,072,495	7,070

TABLE 1: Statistics of the experimental datasets.

5.1.2 Evaluation Metrics

For the explanation generation task, we use BLEU [57], ROUGE [58], and METEOR [59] as the evaluation metrics, which evaluate the text similarity between the generated and gold explanations. BLEU evaluates the n-gram overlap between gold and generated explanations. ROUGE evaluates the recall, precision, and accuracy of the n-gram overlap. METEOR calculates the harmonic mean of each word precision and recall based on the whole corpus. However, these traditional language generation metrics can not measure whether the predicted explanations could express the gold aspects. To better evaluate the generated explanation, we also employ the Feature Matching Ratio (FMR) [14] to measure whether generated explanation can include the target aspects. To evaluate preference prediction performance of different methods, we use Mean Absolute Error (MAE) as the evaluation metric. The definition of MAE is as follows,

$$\mathrm{MAE} = \frac{1}{|\mathcal{T}|} \sum_{(u,i)\in\mathcal{T}} \left|\hat{r}_{ui} - r_{ui}\right|, \tag{23}$$

where $\mathcal{T}$ denotes the set of test data, $\hat{r}_{ui}$ denotes the predicted rating value, r_{ui} denotes the rating value in test data, $|\cdot|$ denotes the size of a set. Note that larger BLEU, ROUGE, METEOR, and FMR values indicate better results for explanation generation task, and lower MAE values indicate better performance for preference prediction task.

5.1.3 Baseline Methods

We compare AESG with the following state-of-the-art explanation generation methods,

- **Att2Seq** [10]: It incorporates the Seq2Seq model [60] and attention mechanism to learn the user's preference from the user attributes and generate review explanations.
- **ExpNet** [12]: It utilizes an encoder-decoder framework to expand a short phrase to a long review by combining the user and item information with other auxiliary side information.
- **Ref2Seq** [13]: This method follows the structure of Seq2Seq and learns the representation from the user and item reviews to generate explanations.
- **NETE-PMI** [14]: It adopts MLP to predict the rating and then generates a template-controlled sentence with the predicted aspect.
- **ACF** [15]: It uses MLP to encode different attributes and applies a coarse-to-fine decoding model to generate long reviews.
- **PETER** [18]: This method uses the Transformer structure for personalized explanation generation. It can simultaneously make recommendation and generates recommendation explanation based on the user and item IDs.

Moreover, we also compare AESG with NETE-PMI, PETER, and the following rating prediction methods to evaluate its ability in predicting users' preferences,

- **PMF** [61]: This is the probabilistic matrix factorization method developed for rating prediction.
- **SVD++** [62]: This method exploits both the user's preferences on items and the influences between items for recommendation.
- **CARL** [63]: This method uses CNNs to learn relevant aspects from the review data;
- **RMG** [37]: It uses a multi-view learning framework to incorporate the review contents and the users' rating behaviors for the recommendation.

5.1.4 Implementation Details

In this work, all the evaluated methods are implemented by PyTorch [64]. For explanation generation methods, the learning rate is chosen from $\{0.0005, 0.0007, 0.002\}$, and the batch size is set to 16. We empirically set the max vocabulary size d_v to 30,000. All the remaining words are replaced by the special token $\langle$UNK$\rangle$. For single-aspect explanation generation, we set the max sentence length to 15. For multi-aspect explanation generation, we set the max feature length to 4 and set the max sentence length to 25. During the inference process, we use the greedy search algorithm to generate explanations.

In AESG, we set the hidden size d_1 and word embedding size to 128. The pre-trained Google News vectors[2] are used to initialize the word embeddings. The graph node embedding size d_0 is chosen from $\{8, 16, 32, 64, 128\}$, and the dimension of user and item id embeddings d_2 is chosen from $\{8, 16, 32, 64, 128\}$. Adam is used to optimize the model with a cosine annealing learning rate decay [65]. The number of graph pooling layers L is set to 2. Moreover, we set the number of top-ranked aspects n extracted from the user/item reviews to 4. In Att2Seq, we set the dimension of attribute embeddings to 64. For the decoding process, the dimension of word embeddings and hidden vectors are set to 512. The dropout rate is set to 0.2. For the ExpNet model, we set the dimension of attribute embeddings and word embedding to 64 and 512 respectively. The dimension of the aspect embedding is 15. For the decoder part, we set the hidden size of GRU to 512, and the dropout rate is set to 0.1. For Ref2Seq, the dimension of word embeddings and hidden vector are 256. For encoder and decoder, the dropout rates are set to 0.5 and 0.2. For NETE-PMI, we set the dimensions of word embeddings and attribute vectors to 200. The size of RNN hidden states is set to 256. The dropout rate is 0.2, and the regularization parameter is set to 0.0001. In ACF, aspects are extracted by TwitterLDA, and the number of aspects is set to 10. For each aspect, the number of keywords is set to 50. The dimension of word embeddings is set to 512. The hidden size of the 2-layer GRU is set to 512.

The hyper-parameters of preference prediction methods are set as follows. For PMF and SVD++, the learning rate is chosen from {0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05}, the dimensionality of embeddings is chosen in {8, 16, 32, 48, 64, 128}, and the batch size is set to 512. For RMG, the learning rate is selected from {0.01, 0.005, 0.001, 0.0005}, the number of CNNs filters is set to 150 and the kernel size is 3. For CARL, the learning rate is chosen from the {0.05, 0.01, 0.005, 0.001}. The dimension of ID embedding and hidden state are 50 and 200. The number of CNN filters is 40.

[2] https://code.google.com/archive/p/word2vec/

Datasets	Methods	Single-aspect Generation						Multi-aspect Generation					
		FMR	B-1 (%)	B-4 (%)	R-1 (%)	R-L (%)	M (%)	FMR	B-1 (%)	B-4 (%)	R-1 (%)	R-L (%)	M (%)
Kindle	Att2Seq	0.22	13.96	1.95	14.96	11.77	5.81	0.46	18.01	2.19	19.80	13.47	6.88
	ExpNet	0.27	14.80	2.86	15.65	12.80	5.95	0.39	12.87	2.13	17.31	11.97	5.09
	Ref2Seq	0.29	15.46	3.07	17.66	14.42	6.68	0.46	13.47	2.29	20.91	14.34	6.27
	NETE-PMI	0.16	12.09	1.21	13.20	10.32	4.96	-	-	-	-	-	-
	ACF	0.13	14.55	3.08	15.58	13.14	5.96	0.18	12.52	2.20	17.43	11.80	5.20
	PETER	**0.49**	16.36	2.91	16.88	13.90	7.05	0.45	13.99	2.49	19.33	13.28	6.20
	AESG	0.43	**16.42**	**3.34**	**18.08**	**14.65**	**7.15**	**0.65**	**19.10**	**2.66**	**23.13**	**14.88**	**8.37**
Elec.	Att2Seq	0.06	11.08	0.43	11.35	8.81	4.25	0.18	14.93	0.67	16.25	11.03	5.29
	ExpNet	0.05	12.03	0.36	12.20	9.51	4.60	0.18	14.64	0.58	16.69	11.45	4.97
	Ref2Seq	0.10	12.70	**0.66**	14.05	**10.99**	4.78	0.21	11.78	0.64	17.82	11.98	4.86
	NETE-PMI	0.11	11.28	0.62	12.08	9.45	4.90	-	-	-	-	-	-
	ACF	0.08	11.31	0.34	12.51	9.57	3.33	0.18	11.99	0.31	14.69	9.38	4.08
	PETER	**0.18**	13.32	0.64	14.07	10.93	**5.50**	0.28	15.04	0.80	17.85	11.98	5.46
	AESG	0.11	**13.84**	0.65	**14.10**	10.90	5.44	**0.31**	**16.78**	**0.86**	**18.54**	**13.34**	**6.97**
Yelp	Att2Seq	0.06	12.06	0.92	12.45	9.84	4.33	0.19	16.70	0.91	17.19	13.00	5.61
	ExpNet	0.08	8.53	0.69	14.34	11.02	4.44	0.22	16.13	1.03	18.23	14.47	5.55
	Ref2Seq	0.09	13.32	**1.05**	16.16	12.87	5.15	0.23	13.45	0.96	18.93	14.67	5.42
	NETE-PMI	0.10	11.73	0.84	14.22	11.05	4.40	-	-	-	-	-	-
	ACF	0.08	12.26	0.54	14.36	11.87	3.74	0.22	14.09	0.65	15.47	11.63	4.08
	PETER	**0.14**	11.66	1.04	15.85	12.30	4.98	**0.31**	13.72	1.03	**19.12**	14.26	5.57
	AESG	0.10	**14.52**	1.04	**16.94**	**13.35**	**5.33**	0.27	**17.07**	**1.07**	18.30	**14.70**	**6.88**

TABLE 2: Explanation generation performance achieved by different methods in terms of FMR, BLEU (%), ROUGE (%), METEOR (%). B, R and M refer to BLEU, ROUGE and METEOR. Note that NETE-PMI generates explanations with a single aspect, thus we only report its single-aspect explanation generation performance.

Datasets	Methods	Single-aspect Generation						Multi-aspect Generation					
		FMR	B-1 (%)	B-4 (%)	R-1 (%)	R-L (%)	M (%)	FMR	B-1 (%)	B-4 (%)	R-1 (%)	R-L (%)	M (%)
Kindle	PETER	**0.49**	16.47	3.04	17.20	14.18	7.01	0.45	14.04	2.45	19.53	13.50	6.21
	AESG	0.45	**16.59**	**3.36***	**18.32***	**14.87****	**7.14****	**0.63***	**19.56***	**2.61***	**23.25***	**15.10***	**8.48***
Elec.	PETER	**0.19**	13.53	**0.69**	14.41	11.08	5.55	0.29	14.94	0.77	17.78	11.93	5.41
	AESG	0.11	**14.56***	**0.69**	**15.13***	**11.50***	**5.73****	**0.32***	**16.86***	**0.87***	**19.06***	**12.60***	**7.14***
Yelp	PETER	**0.14**	11.95	**1.08**	15.95	12.41	5.00	**0.31**	13.73	1.03	**19.16**	14.25	5.59
	AESG	0.09	**14.39***	1.02	**16.69***	**13.26***	**5.45***	0.25	**17.12***	**1.12***	18.36	**14.61***	**6.83***

TABLE 3: The average explanation generation performance achieved by AESG and the best baseline method PETER, after repeating the experiments five times. * and ** indicate the statistical significance over the best baseline respectively for p<0.01 and p<0.05 via Student's t-test.

Methods	Kindle	Electronics	Yelp
PMF	0.6214	0.8292	0.7982
SVD++	0.5723	0.8492	0.8022
CARL	0.5223	**0.7794**	0.7766
RMG	0.5555	0.8203	0.8035
NETE-PMI	0.5284	0.8537	0.7763
PETER	0.7939	0.9938	0.9290
AESG	**0.5069**	0.7836	**0.7760**

TABLE 4: The preference prediction performance achieved by different methods in terms of MAE. The best results are in **bold** faces and the second-best results are underlined.

5.2 Performance Comparison

Table 2 summarizes the performance of the single-aspect and multi-aspect explanation generation tasks achieved by different methods. As shown in Table 2, ExpNet usually achieves better performance than Att2Seq by using aspects to guide the generation process. Ref2Seq obtains better results than ExpNet and Att2Seq, especially on FMR metric. One potential reason is that Ref2Seq uses the user's historical reviews as inputs for explanation generation. And the review data contains more aspect-relevant information that can help include more aspects in the generated sentence. In addition, ExpNet outperforms NETE-PMI on the Kindle dataset, in terms of all metrics. It may because that ExpNet uses an attention fusion layer to control the generation outputs. Compared with Ref2Seq, ACF does not extract side information from the review data as input, and it also generates sentences by filling the generated templates based on predicted aspects. Its explanation generation performance is highly dependent on the quality of input aspects. In ACF, the aspects are extracted by TwitterLDA [66], thus the aspect quality could be influenced by the review data quality and the given number of latent topics. We also observe that PETER achieves better performance than Ref2seq, especially in the multi-aspect generation task. One potential reason is that PETER employs an extra task to support text generation.

Moreover, as shown in Table 2, AESG usually achieves the best explanation generation performance on all datasets, by enhancing explanation generation with user and item review-based syntax graphs. The review-based syntax graph aggregates information from all the reviews associated with the user/item to provide a unified view of the aspect-relevant details about the user/item. Thus, the review-based syntax graph can also help weaken the impacts caused by noise reviews. To further investigate the improvements achieved by AESG, we first repeat the experiments of AESG and the best baseline method PETER five times. Table 3 summarizes the average explanation generation performance achieved by PETER and AESG. We can note that AESG consistently outperforms PETER on all three datasets. Then, we perform Student's t-test over the results of PETER and AESG. As shown in Table 3, most of the improvements achieved by AESG are statistically significant with p-value smaller than 0.05.

Methods	FMR	BLUE-4 (%)	ROUGE-L (%)	MAE
AESG	0.65	**2.66**	14.88	**0.5069**
AESG$_{\text{w/o AGP}}$	**0.68**	2.39	14.40	0.5075
AESG$_{\text{GAT}}$	0.65	2.33	14.41	0.5099
AESG$_{\text{LSTM}}$	0.63	2.43	14.71	0.5127
AESG$_{\text{BERT}}$	0.62	2.58	15.47	0.5339
AESG$_{\text{MLP}}$	0.57	2.41	12.24	0.5095
AESG$_{\text{implicit}}$	0.64	2.22	14.90	0.5084
AESG$_{\text{w/o Relation}}$	0.57	2.58	**15.50**	0.5141

TABLE 5: The performance achieved by different variants of AESG on the Kindle dataset.

The preference prediction performance achieved by different methods is shown in Table 4. We can note that the review-based methods usually achieve better performance than traditional matrix factorization method (*i.e.*, PMF and SVD++). Moreover, the proposed AESG method achieves the best preference prediction performance on Kindle and Yelp datasets, and achieves the second best prediction performance on Electronics dataset, in terms of MAE. These observations indicate that the user preference predicted by AESG can support the high-quality explanation generation.

5.3 Ablation Study

In this section, we perform ablation study to analyze the effectiveness of different components of AESG.

5.3.1 Impacts of AGP Operator

To study the contributions of the AGP operator, we study the performance of the following three variants of AESG,

- **AESG**$_{\text{w/o AGP}}$: In this variant, the AGP operator is removed from the model. For the aspect and explanation attention, we use the node feature of the original review-based syntax graph to replace the node feature of the L-th graph.
- **AESG**$_{\text{GAT}}$: In this variant, the aspect-aware graph pooling module of AGP is removed. We only use GAT to learn the representation of the review-based syntax graph.
- **AESG**$_{\text{LSTM}}$: In this variant, the AGP operator is replaced with LSTM to update the representations of nodes in the graph. For the aspect and explanation attention calculation, it is same as AESG$_{\text{w/o AGP}}$.
- **AESG**$_{\text{BERT}}$: In this variant, the AGP operator is replaced with the pre-trained BERT model, and other settings follow AESG$_{\text{LSTM}}$.

As shown in Table 5, AESG outperforms AESG$_{\text{w/o AGP}}$, AESG$_{\text{GAT}}$, and AESG$_{\text{LSTM}}$, in terms of BLEU-4, ROUGE-L, and MAE. This demonstrates that the AGP operator can benefit both the prediction of user preference and the explanation generation. Besides, AESG$_{\text{BERT}}$ achieves better performance than AESG$_{\text{LSTM}}$ in terms of language metrics (*i.e.*, BLUE-4 and ROUGE-L). This indicates that the pre-trained BERT model has better language generation ability than the traditional language model LSTM. Moreover, the performance of AESG and AESG$_{\text{BERT}}$ is comparable in terms of language generation metrics. However, AESG performs better than AESG$_{\text{BERT}}$ in the rating prediction and aspect generation tasks.

The FMR metric measures whether the target aspect can be included in the generated sentence. As shown in Table 5, AESG outperforms most variants, except AESG$_{\text{w/o AGP}}$ and AESG$_{\text{GAT}}$, in terms of FMR. Specifically, the AESG can achieve 0.65 in terms of FMR which means that 65% of generated sentences can include the target aspect. Although AESG$_{\text{w/o AGP}}$ achieves a better FMR value than AESG, it achieves worse performance in BLUE-4 and ROUGE-L. One potential reason is that AESG$_{\text{w/o AGP}}$ only considers the word-level (*i.e.*, node-level) information in the review-based syntax graph, which is beneficial for aspect generation. However, AESG employs graph pooling to obtain hierarchical representations of the review-based syntax graph, which can exploit both the world-level information and other high-level information, *e.g.*, sentence-level information, from the review-based syntax graph. This high-level information can be described by substructures of the review-based syntax graph, and it is beneficial for sentence explanation generation. As AESG captures more high-level information, it may ignore some world-level information. Thus, its FMR value is lower than that of AESG$_{\text{w/o AGP}}$.

ρ	FMR	BLEU-4 (%)	ROUGE-L (%)	MAE
0.1	**0.66**	2.25	14.13	0.5107
0.3	0.53	2.57	14.67	**0.5069**
0.5	0.65	**2.66**	14.88	**0.5069**
0.7	**0.66**	2.49	14.68	0.5159
0.9	0.52	2.54	**14.99**	0.5126

TABLE 6: Performance of AESG with respect to (w.r.t.) different graph pooling ratio ρ on the Kindle dataset.

L	FMR	BLEU-4 (%)	ROUGE-L (%)	MAE
1	**0.69**	2.22	13.78	0.5144
2	0.65	**2.66**	14.88	**0.5069**
3	0.65	2.62	**15.79**	0.5081
4	0.68	2.26	13.70	0.5113

TABLE 7: Performance of AESG w.r.t. different settings of the number of pooling layers L on the Kindle dataset.

5.3.2 Impacts of Historical Aspects

To study the impacts of historical aspects, we consider the following two variants of AESG for evaluation,

- **AESG**$_{\text{MLP}}$: In this variant, we replace BiLSTM with MLP to extract features from the historical aspects.
- **AESG**$_{\text{Implicit}}$: In this variant, we replace the feature of the historical aspect with a randomly initialized vector, which can be learned in model training.

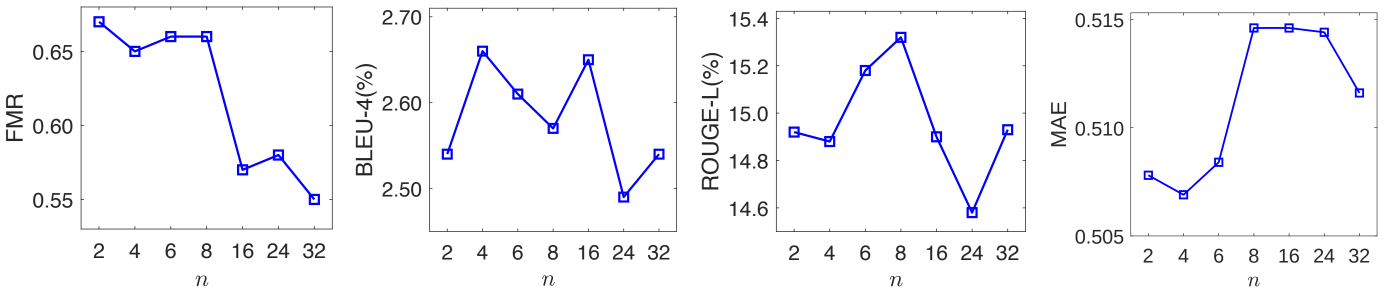
As shown in Table 5, AESG achieves better performance than AESG$_{\text{MLP}}$ and AESG$_{\text{Implicit}}$. This observation indicates that the historical aspect set is helpful to mine the user preference for aspects, and the BiLSTM structure can help learn better representations for historical aspects.

5.3.3 Impacts of Grammatical Relations

To study the impacts of grammatical relations in the review-based syntax graph, we remove the relation embedding in Eq. (3) and denote this variant of AESG by AESG$_{\text{w/o Relation}}$. As shown in Table 5, AESG achieves better FMR, BLUE-4 and MAE values than AESG$_{\text{w/o Relation}}$. This indicates that the grammatical relations can help improve the explanation generation performance.

5.4 Parameter Sensitivity Study

In this work, we employ the AGP operator to mine the aspect-relevant information from the review-based syntax



(a) FMR (b) BLEU-4 (c) ROUGE-L (d) MAE

Fig. 4: Performance of AESG with respect to different settings of historical aspect number n on the Kindle dataset.

Model	FMR	BLEU-4 (%)	ROUGE-L (%)	MAE
AESG$_{\mathrm{B}}$	0.63	**2.94**	**15.38**	0.5216
AESG$_{\mathrm{G}}$	0.65	2.66	14.88	**0.5069**
AESG$_{\mathrm{w/o\ P}}$	**0.67**	2.60	15.30	0.5102
Ref2Seq$_{\mathrm{B}}$	0.31	2.51	12.73	-
Ref2Seq$_{\mathrm{G}}$	0.46	2.29	14.34	-
Ref2Seq$_{\mathrm{w/o\ P}}$	0.45	2.22	14.22	-

TABLE 8: Performance of AESG and Ref2Seq with/without pre-trained word embeddings on Kindle dataset. "B" and "G" means using BERT or GloVe as pre-trained embeddings, and "w/o P" means without pre-trained embedding. Note Ref2Seq does not include the preference prediction module.

graph. We perform experiments to evaluate the impacts of the number of historical aspects n on the performance of AESG. Figure 4 shows performance trends of AESG with respect to different settings of n. As shown in Figure 4, we can note that AESG achieves the best FMR value when n is set to 2. This indicates that the generated sentences have the highest probability to cover the user's interested aspects. The best BLEU-4 and MAE values are achieved when n is set to 4. Moreover, the best ROUGE-L value is achieved when n is set to 8. In the experiments, we empirically set n to 4, due to its best performance on BLEU-4.

Moreover, we also study the performance of AESG with respect to different settings of the graph pooling ratio ρ, which represents the proportion of nodes retained. When ρ is set to 0.1, it means that the model only retains 10% of nodes in each graph pooling operation. As we set the graph layer to 2, only 1% of nodes in the review-based syntax graph are remained after two AGP operations. Table 6 summarizes the performance of AESG with respect to different settings of ρ. We can note that the best FMR values are achieved by setting ρ to 0.1 and 0.7. For text generation metrics, the best BLEU-4 and ROUGE-L values are achieved when ρ is set to 0.5 and 0.9, respectively. The best preference prediction performance in terms of MAE is achieved when ρ is set to 0.3 and 0.5.

Table 7 shows the performance of AESG with respect to different settings of the number of pooling layers L on the Kindle dataset. As shown in Table 7, the best FMR value is achieved when L is set to 1, and better BLEU-4, ROUGE-L, and MAE values are obtained when L is set to 2 and 3. When L is set to 1, the AESG model only aggregates the first-order neighbourhood information of nodes in the review-based syntax graph. AESG mainly captures the word-level information that can benefit aspect generation. Thus, a better FMR value is achieved when L is 1. When L is set to 2 or 3, AESG can capture more high-order neighbourhood information of nodes in the syntax graph, which can be treated as the sentence-level textual information that can help the sentence explanation generation. Thus, AESG achieves better performance in BLEU-4, and ROUGE-L, when L is set to 2 and 3. When $L = 2$, AESG can achieve the best performance in terms of BLEU-4 and MAE, thus it is appropriate to set L to 2 in the experiments.

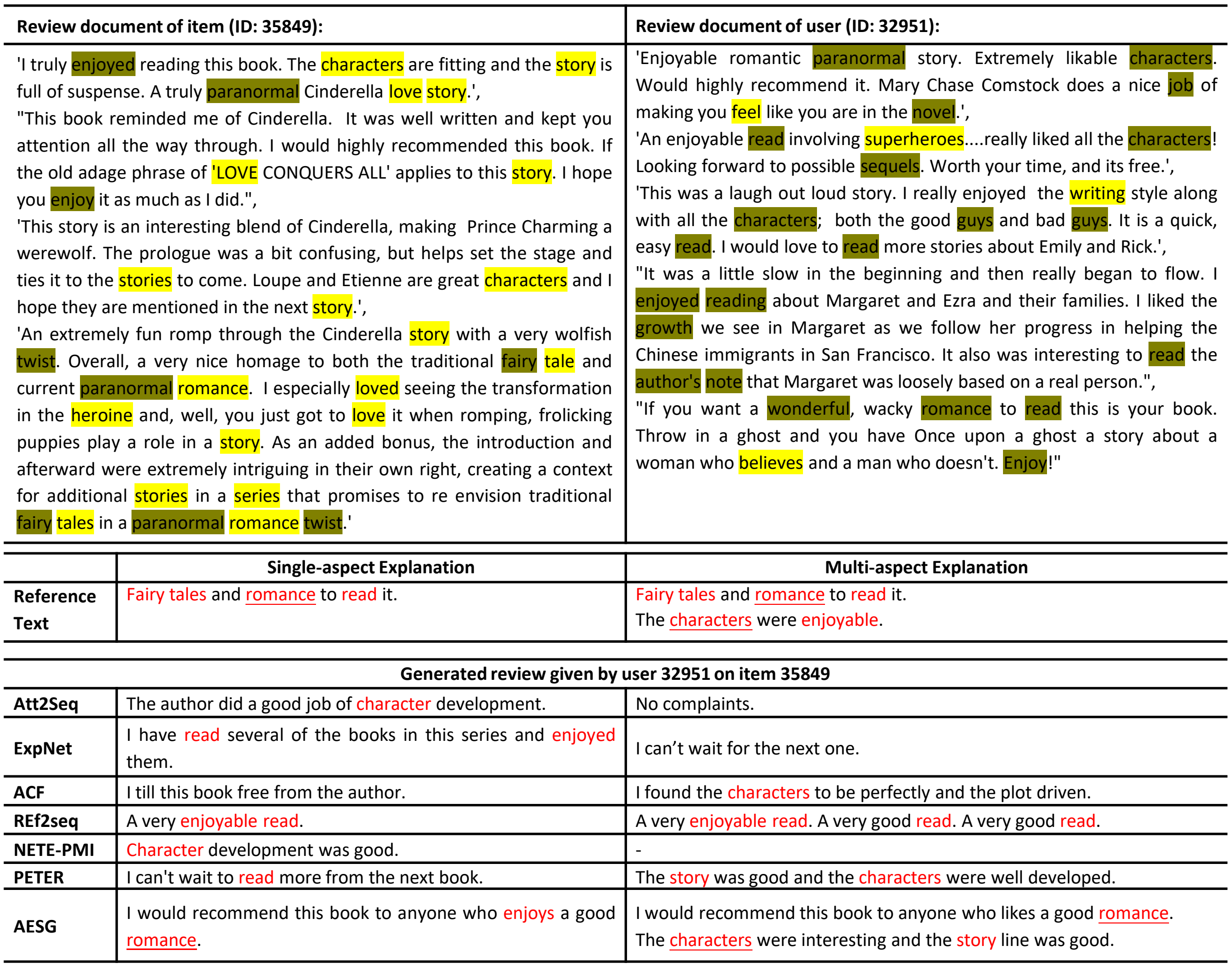
5.5 Impacts of Pre-trained Embeddings

Pre-trained embeddings are obtained from models trained on large datasets, thus they usually contain rich semantic information. To evaluate the performance of AESG without pre-trained embeddings and the effectiveness of pre-trained embeddings in our tasks, we summarize the performance of AESG with/without pre-trained embeddings on the Kindle dataset in Table 8. Here, we only compare AESG with Ref2Seq, which is the best baseline method in explanation generation, and it also uses reviews as inputs.

As shown in Table 8, the pre-trained BERT embeddings can help improve the performance of both models in terms of BLEU-4. Moreover, AESG$_{\mathrm{w/o\ P}}$ achieves better FMR value than AESG$_{\mathrm{B}}$ and AESG$_{\mathrm{G}}$. This indicates that using pre-trained embeddings cannot help the proposed AESG model cover more user-interested aspects in the generated sentences. Moreover, we can also note that the proposed AESG model consistently outperforms baseline method Ref2Seq under different settings.

5.6 Case Study

Figure 5 shows examples of the single-aspect and multi-aspect explanations generated by different methods for a given user-item pair. We highlight the important words in the review documents based on the word importance (refer to Eq.(4) for definition) extracted from the first graph layer of the AGP operator. As shown in Figure 5, "fairy", "tales", "romance", "character", "story", "love", and "enjoy" are considered important words and selected in the first pooling layer of AESG. Most of these important words appear in the reference text, indicating the proposed AESG model can capture important information from the input data. Compared with the explanations generated by baseline methods, the generated sentences from AESG cover more important words, *e.g.*, "enjoy", "romance", "characters", and "story". This indicates that the selected words can help generate



Review document of item (ID: 35849):	Review document of user (ID: 32951):
'I truly enjoyed reading this book. The characters are fitting and the story is full of suspense. A truly paranormal Cinderella love story.', "This book reminded me of Cinderella. It was well written and kept you attention all the way through. I would highly recommended this book. If the old adage phrase of 'LOVE CONQUERS ALL' applies to this story. I hope you enjoy it as much as I did.", 'This story is an interesting blend of Cinderella, making Prince Charming a werewolf. The prologue was a bit confusing, but helps set the stage and ties it to the stories to come. Loupe and Etienne are great characters and I hope they are mentioned in the next story.', 'An extremely fun romp through the Cinderella story with a very wolfish twist. Overall, a very nice homage to both the traditional fairy tale and current paranormal romance. I especially loved seeing the transformation in the heroine and, well, you just got to love it when romping, frolicking puppies play a role in a story. As an added bonus, the introduction and afterward were extremely intriguing in their own right, creating a context for additional stories in a series that promises to re envision traditional fairy tales in a paranormal romance twist.'	'Enjoyable romantic paranormal story. Extremely likable characters. Would highly recommend it. Mary Chase Comstock does a nice job of making you feel like you are in the novel.', 'An enjoyable read involving superheroes....really liked all the characters! Looking forward to possible sequels. Worth your time, and its free.', 'This was a laugh out loud story. I really enjoyed the writing style along with all the characters; both the good guys and bad guys. It is a quick, easy read. I would love to read more stories about Emily and Rick.', "It was a little slow in the beginning and then really began to flow. I enjoyed reading about Margaret and Ezra and their families. I liked the growth we see in Margaret as we follow her progress in helping the Chinese immigrants in San Francisco. It also was interesting to read the author's note that Margaret was loosely based on a real person.", "If you want a wonderful, wacky romance to read this is your book. Throw in a ghost and you have Once upon a ghost a story about a woman who believes and a man who doesn't. Enjoy!"

	Single-aspect Explanation	Multi-aspect Explanation
Reference Text	Fairy tales and romance to read it.	Fairy tales and romance to read it. The characters were enjoyable.

	Generated review given by user 32951 on item 35849	
Att2Seq	The author did a good job of character development.	No complaints.
ExpNet	I have read several of the books in this series and enjoyed them.	I can't wait for the next one.
ACF	I till this book free from the author.	I found the characters to be perfectly and the plot driven.
REf2seq	A very enjoyable read.	A very enjoyable read. A very good read. A very good read.
NETE-PMI	Character development was good.	-
PETER	I can't wait to read more from the next book.	The story was good and the characters were well developed.
AESG	I would recommend this book to anyone who enjoys a good romance.	I would recommend this book to anyone who likes a good romance. The characters were interesting and the story line was good.

Fig. 5: Explanations generated by different methods for a given user-item pair. The highlighted words in review documents denote the key information captured by AESG. The words with dark yellow background have larger importance scores than the words with a yellow background. The words in red color match the important words in the review documents. The target aspects are underlined. Note that NETE-PMI cannot generate multi-aspect relevant explanations.

high-quality explanations. Moreover, for single-aspect generation, ”romance” is the target aspect. In the multi-aspect generation task, ”romance” and ”character” are the target aspects. We can see that the explanations generated by AESG can better express these target aspects than baseline methods. In addition, the explanation sentences generated by AESG are also more natural than those generated by baseline methods.

Moreover, Table 9 shows explanations generated for different users, with the same given item. We can note different users can receive different explanations even for the same item. The generated explanation sentences can cover the different important information for different user-item pairs, *e.g.*, “book”, “illustrations”, and “cute”. To have a better understanding of personalized explanations, we study the overlap of generated aspects for different users, with the same given item. Specifically, we define the following aspect overlapping ratio (AOR) for testing items,

$$\mathrm{AOR}(i) = \frac{1}{|\mathcal{U}_i^T|(|\mathcal{U}_i^T|-1)} \sum_{u\in\mathcal{U}_i^T} \sum_{u'\in\mathcal{U}_i^T, u'\neq u} \frac{|\mathcal{A}_{ui}\cap\mathcal{A}_{u'i}|}{|\mathcal{A}_{ui}\cup\mathcal{A}_{u'i}|},$$

$$\mathrm{AOR} = \frac{1}{|\mathcal{I}_T|} \sum_{i\in\mathcal{I}_T} \mathrm{AOR}(i), \tag{24}$$

where $\mathcal{U}_i^T$ denotes the set of users that have interactions with i in the testing data, $\mathcal{A}_{ui}$ denotes the set of aspects generated for the testing (u, i) pair, and $\mathcal{I}_T$ denotes the set of items in testing data. Table 10 summarizes the AOR values and the average number of aspects (ANA) of the explanations generated by AESG on three datasets. The results in Table 10 further demonstrate that the proposed AESG method can generate personalized explanations for different users even on the same item. Note that the ANA values in single-aspect generation task are larger than 1. This is because that, besides the generated aspects, the explanation decoder of AESG may generate some other words that are also aspects.

6 Conclusion

This paper proposes a novel explainable recommendation model, namely Aspect-guided Explanation generation with Syntax Graph (AESG). Specifically, AESG employs a review-based syntax graph that captures the user/item details from the review data to enhance explanation generation. An aspect-guided graph pooling (AGP) operator is proposed to distill the aspect-relevant information from the review-based syntax graph. Moreover, an aspect matching

Item ID	User ID		Explanations
35726	33892	Reference Text AESG	Terry treetop is a delightful series of children's books. This is a great book for kids to read.
	24091	Reference Text AESG	Where is my home is an entertaining and fun story with colourful illustrations. The illustrations are great and the story is very well written to read.
	32318	Reference Text AESG	The colourful illustrations and the cute characters make this book a good read for young. This is a cute book this book is a great little read.
31066	91648	Reference Text AESG	Been using these for years and they are great for the price. I have had these discs for years and they are great time thing I use for a few years.
	29221	Reference Text AESG	These DVD s work fine and seem to be just as good as other brands. The quality is very good and the price is right.
	98885	Reference Text AESG	Good price these were a good price so I bought these. The price is right I have been using these for a few years and they are very good quality thing.

TABLE 9: Explanations generated by AESG for different users, with the same given item. The highlighted word appears in both ground-truth and generated sentences.

Dataset	Single-aspect Generation		Multi-aspect Generation	
	AOR	ANA	AOR	ANA
Kindle	0.2392	2.8543	0.1407	6.7743
Elec.	0.1433	2.1300	0.1118	6.4207
Yelp	0.3489	1.8843	0.3587	7.2043

TABLE 10: The aspect overlapping ratio (AOR) and the average number of aspects (ANA) of the explanations generated by AESG on three datasets.

mechanism is developed to match the user preferences and item properties at the aspect level. Furthermore, an aspect-guided decoder is also developed to first predict the user interested aspects and then generate the aspect-relevant explanations. The experimental results on real datasets demonstrate that AESG outperforms state-of-the-art explanation generation methods.

7 Acknowledgments

This research is supported, in part, by the National Research Foundation (NRF), Singapore under its AI Singapore Programme (AISG Award No: AISG-GC-2019-003), and by the Development of Cryptographic Library and Support Systems (LP180101062), Australian Research Council. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore and Australian Research Council.

References

[1] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," *ACM Computing Surveys (CSUR)*, vol. 47, no. 1, pp. 1–45, 2014.

[2] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.

[3] R. Catherine and W. Cohen, "Transnets: Learning to transform for recommendation," in *Proceedings of the eleventh ACM conference on recommender systems*, 2017, pp. 288–296.

[4] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Neural attentional rating regression with review-level explanations," in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 1583–1592.

[5] H. Chen, X. Chen, S. Shi, and Y. Zhang, "Generate natural language explanations for recommendation," in *Proceedings of SIGIR'19 Workshop on ExplainAble Recommendation and Search. ACM*, 2019.

[6] X. Chen, H. Chen, H. Xu, Y. Zhang, Y. Cao, Z. Qin, and H. Zha, "Personalized fashion recommendation with visual explanations based on multimodal attention network: Towards visually explainable recommendation," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 765–774.

[7] P. Sun, L. Wu, K. Zhang, Y. Fu, R. Hong, and M. Wang, "Dual learning for explainable recommendation: Towards unifying user preference prediction and review generation," in *Proceedings of The Web Conference 2020*, 2020, pp. 837–847.

[8] Y. Zhang, X. Chen *et al.*, "Explainable recommendation: A survey and new perspectives," *Foundations and Trends® in Information Retrieval*, vol. 14, no. 1, pp. 1–101, 2020.

[9] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma, "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, 2014, pp. 83–92.

[10] L. Dong, S. Huang, F. Wei, M. Lapata, M. Zhou, and K. Xu, "Learning to generate product reviews from attributes," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 2017, pp. 623–632.

[11] P. Li, Z. Wang, Z. Ren, L. Bing, and W. Lam, "Neural rating regression with abstractive tips generation for recommendation," in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2017, pp. 345–354.

[12] J. Ni and J. McAuley, "Personalized review generation by expanding phrases and attending on aspect-aware representations," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018, pp. 706–711.

[13] J. Ni, J. Li, and J. McAuley, "Justifying recommendations using distantly-labeled reviews and fine-grained aspects," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 188–197.

[14] L. Li, Y. Zhang, and L. Chen, "Generate neural template explanations for recommendation," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 755–764.

[15] J. Li, W. X. Zhao, J.-R. Wen, and Y. Song, "Generating long and informative reviews with aspect-aware coarse-to-fine decoding," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 1969–1979.

[16] Y. Xian, T. Zhao, J. Li, J. Chan, A. Kan, J. Ma, X. L. Dong, C. Faloutsos, G. Karypis, S. Muthukrishnan *et al.*, "Ex3: Explainable attribute-aware item-set recommendations," in *Fifteenth ACM Conference on Recommender Systems*, 2021, pp. 484–494.

[17] N. Wang, H. Wang, Y. Jia, and Y. Yin, "Explainable recommendation via multi-task learning in opinionated text data," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 165–174.

[18] L. Li, Y. Zhang, and L. Chen, "Personalized transformer for explainable recommendation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 4947–4957.

[19] Y. Lin, P. Ren, Z. Chen, Z. Ren, J. Ma, and M. De Rijke, "Explainable outfit recommendation with joint outfit matching and comment

generation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 8, pp. 1502–1516, 2019.

[20] P. Tangseng and T. Okatani, "Toward explainable fashion recommendation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 2153–2162.

[21] Q. Wu, P. Zhao, and Z. Cui, "Visual and textual jointly enhanced interpretable fashion recommendation," *IEEE Access*, vol. 8, pp. 68 736–68 746, 2020.

[22] H. Park, H. Jeon, J. Kim, B. Ahn, and U. Kang, "Uniwalk: Explainable and accurate recommendation for rating and network data," *arXiv preprint arXiv:1710.07134*, 2017.

[23] Z. Ren, S. Liang, P. Li, S. Wang, and M. de Rijke, "Social collaborative viewpoint regression with explainable recommendations," in *Proceedings of the tenth ACM international conference on web search and data mining*, 2017, pp. 485–494.

[24] J. Tang, Y. Yang, S. Carton, M. Zhang, and Q. Mei, "Context-aware natural language generation with recurrent neural networks," *arXiv preprint arXiv:1611.09900*, 2016.

[25] P. Li and A. Tuzhilin, "Towards controllable and personalized review generation," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3237–3245.

[26] P. Sun, L. Wu, K. Zhang, Y. Su, and M. Wang, "An unsupervised aspect-aware recommendation model with explanation text generation," *ACM Transactions on Information Systems (TOIS)*, vol. 40, no. 3, pp. 1–29, 2021.

[27] J. Li, S. Li, W. X. Zhao, G. He, Z. Wei, N. J. Yuan, and J.-R. Wen, "Knowledge-enhanced personalized review generation with capsule graph neural network," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 735–744.

[28] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proceedings of the tenth ACM international conference on web search and data mining*, 2017, pp. 425–434.

[29] D. Liu, J. Li, B. Du, J. Chang, and R. Gao, "Daml: Dual attention mutual learning between ratings and reviews for item recommendation," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 344–352.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[31] Z. Liang, J. Du, and C. Li, "Abstractive social media text summarization using selective reinforced seq2seq attention model," *Neurocomputing*, vol. 410, pp. 432–440, 2020.

[32] Y. Liu, S. Yang, Y. Zhang, C. Miao, Z. Nie, and J. Zhang, "Learning hierarchical review graph representations for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[33] S. Seo, J. Huang, H. Yang, and Y. Liu, "Interpretable convolutional neural networks with dual local and global attention for review rating prediction," in *Proceedings of the eleventh ACM conference on recommender systems*, 2017, pp. 297–305.

[34] L. Wu, X. He, X. Wang, K. Zhang, and M. Wang, "A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[35] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 974–983.

[36] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 2019, pp. 165–174.

[37] C. Wu, F. Wu, T. Qi, S. Ge, Y. Huang, and X. Xie, "Reviews meet graphs: Enhancing user and item representations for recommendation with hierarchical attentive graph neural network," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 4886–4895.

[38] J. Gao, Y. Lin, Y. Wang, X. Wang, Z. Yang, Y. He, and X. Chu, "Set-sequence-graph: A multi-view approach towards exploiting reviews for recommendation," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 395–404.

[39] J. Shuai, K. Zhang, L. Wu, P. Sun, R. Hong, M. Wang, and Y. Li, "A review-aware graph contrastive learning framework for recommendation," in *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, 2022.

[40] Y. Zhang, Y. Liu, Y. Xu, H. Xiong, C. Lei, W. He, L. Cui, and C. Miao, "Enhancing sequential recommendation with graph contrastive learning," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2022, pp. 2398–2405.

[41] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 346–353.

[42] Z. Wang, W. Wei, G. Cong, X.-L. Li, X.-L. Mao, and M. Qiu, "Global context enhanced graph neural networks for session-based recommendation," in *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, 2020, pp. 169–178.

[43] S. Kübler, R. McDonald, and J. Nivre, "Dependency parsing," *Synthesis lectures on human language technologies*, vol. 1, no. 1, pp. 1–127, 2009.

[44] D. Jurafsky and J. H. Martin, "Speech and language processing (draft)," *Chapter A: Hidden Markov Models (Draft of September 11, 2018). Retrieved March*, vol. 19, p. 2019, 2018.

[45] D. Wang, P. Liu, Y. Zheng, X. Qiu, and X.-J. Huang, "Heterogeneous graph neural networks for extractive document summarization," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 6209–6219.

[46] L. Pan, Y. Xie, Y. Feng, T.-S. Chua, and M.-Y. Kan, "Semantic graphs for generating deep questions," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 1463–1475.

[47] X. He, T. Chen, M.-Y. Kan, and X. Chen, "Trirank: Review-aware explainable recommendation by modeling aspects," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015, pp. 1661–1670.

[48] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations*, 2018.

[49] H. Gao and S. Ji, "Graph u-nets," in *international conference on machine learning*. PMLR, 2019, pp. 2083–2092.

[50] H. Gao, Y. Chen, and S. Ji, "Learning graph pooling and hybrid convolutional operations for text representations," in *The World Wide Web Conference*, 2019, pp. 2743–2749.

[51] J. Lee, I. Lee, and J. Kang, "Self-attention graph pooling," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3734–3743.

[52] J. Y. Chin, K. Zhao, S. Joty, and G. Cong, "Anr: Aspect-based neural recommender," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 147–156.

[53] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016, pp. 289–297.

[54] S. Rendle, "Factorization machines," in *2010 IEEE International Conference on Data Mining*. IEEE, 2010, pp. 995–1000.

[55] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *proceedings of the 25th international conference on world wide web*, 2016, pp. 507–517.

[56] Z. Cheng, Y. Ding, L. Zhu, and M. Kankanhalli, "Aspect-aware latent factor model: Rating prediction with ratings and reviews," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 639–648.

[57] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[58] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.

[59] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation*

measures for machine translation and/or summarization, 2005, pp. 65–72.

[60] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[61] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Advances in neural information processing systems*, 2008, pp. 1257–1264.

[62] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 426–434.

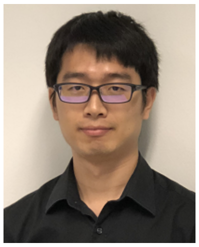
[63] L. Wu, C. Quan, C. Li, Q. Wang, B. Zheng, and X. Luo, "A context-aware user-item representation learning for item recommendation," *ACM Transactions on Information Systems (TOIS)*, vol. 37, no. 2, pp. 1–29, 2019.

[64] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, 2019, pp. 8024–8035.

[65] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, "Snapshot ensembles: Train 1, get m for free," in *International Conference on Learning Representations*, 2017.

[66] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li, "Comparing twitter and traditional media using topic models," in *European conference on information retrieval*. Springer, 2011, pp. 338–349.

Yidan Hu received the B.Eng. degree from Shandong University, Jinan, China, in 2019. She is currently pursuing a Ph.D. degree in the School of Computer Science and Engineering with the University of Nanyang Technological University, Singapore. Her research interests include the recommendation system and Natural Language Processing.



Yong Liu is a Senior Research Scientist at Alibaba-NTU Singapore Joint Research Institute and Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), Nanyang Technological University, Singapore. Before that, he was a Data Scientist at NTUC Enterprise, Singapore from November 2017 to July 2018, and a Research Scientist at Data Analytics Department, Institute for Infocomm Research (I2R), A*STAR, Singapore from November 2015 to October 2017. He received his Ph.D. from the School of Computer Science and Engineering at Nanyang Technological University in 2016 and B.S. from the Department of Electronic Science and Technology at University of Science and Technology of China in 2008. His research interests include recommender systems, natural language processing, and knowledge graph. He has been invited as a PC member of major conferences such as KDD, SIGIR, ACL, IJCAI, AAAI, and reviewer for IEEE/ACM transactions.

Chunyan Miao is a President's Chair Professor and the Chair of the School of Computer Science and Engineering (SCSE) at NTU Singapore. She received her PhD degree in Computer Engineering from NTU and was an NSERC Postdoctoral Fellow at Simon Fraser University (SFU), Canada. She was a founding faculty member of the Centre for Digital Media established by The University of British Columbia (UBC) and SFU. She was also a Tan Chin Tuan Engineering Fellow at Harvard and MIT. Dr. Miao has received over 20 Best Paper/innovation awards in Artificial intelligence (AI) and real-world AI applications for her impactful research in health, ageing, education and smart services. She is a recipient of the prestigious NRF Investigatorship Award 2018. She also holds major research funding including MOH National Innovation Challenge (NIC) on Ageing award 2018 and NRF AI.SG Health Grand Challenge Award 2019. She is the Founding Director of the Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), Singapore's first centre focusing on AI empowered solutions to population aging challenges. She is also the Founding Director of the Alibaba-NTU Singapore Joint Research Institute (JRI), Alibaba's first and largest JRI outside China. She is an Editor/Associate Editor of leading international journals including IJIT, IEEE Big Data, IEEE IoT, IEEE Access and IEEE Service Computing and has served as Chair/TPC member of international conferences such as IEEE ICA, ICAA, ACM KDD. She serves on various national committees, including the MOH City for All Ages and Health Tech, the IMDA TechSkills Accelerator (TeSA) and is the Chair of the SCS AI Ethics Review Committee. She was awarded a Public Administration Medal (Bronze) from the President of Singapore in 2016.

Gongqi Lin is a lecturer of Information Technology with the College of Engineering & Science. He received a PhD degree in Computing from Curtin University, Australia, in 2014. Prior to joining VU, Gongqi worked as a Research Associate at Curtin University. He also worked as a Senior Software Engineer in the IT industry. His research area includes natural language processing, knowledge graph and software engineering.

Yuan Miao received his PhD from Tsinghua University 1996. He is currently the Head of IT Discipline at the College of Engineering and Science, Victoria University, leading the Innovative Intelligent Technology Lab research. His research interests are in human knowledge modelling (fuzzy cognitive map, knowledge graph), natural language comprehension, decision support and their application in smart cities, health care, active aging, cyber security, and digital transformation in the construction industry. He has led the research towards the discovery of the sufficient and necessary condition for a complex cognitive map to be decomposable, a transformation of major cognitive map models, an adversary dataset that fails BERT and ELECTRA. His research received support from the industry, including Oracle, Amazon, Microsoft, and national competitive grant support from government funding bodies such as Australia Research Council and National Research Foundation Singapore.