Knowledge-based Review Generation by Coherence Enhanced Text Planning

Junyi Li1,2, Wayne Xin Zhao1,2, Zhicheng Wei3, Nicholas Jing Yuan3 and Ji-Rong Wen1,2
1Gaoling School of Artificial Intelligence, Renmin University of China
2Beijing Key Laboratory of Big Data Management and Analysis Methods
3Huawei Cloud
{ljunyi,jrwen}@ruc.edu.cn, {batmanfly, nicholas.jing.yuan}@gmail.com, weizhicheng1@huawei.com

ABSTRACT
As a natural language generation task, it is challenging to generate informative and coherent review text. In order to enhance the informativeness of the generated text, existing solutions typically learn to copy entities or triples from knowledge graphs (KGs). However, they lack overall consideration to select and arrange the incorporated knowledge, which tends to cause text incoherence.

To address the above issue, we focus on improving entity-centric coherence of the generated reviews by leveraging the semantic structure of KGs. In this paper, we propose a novel Coherence Enhanced Text Planning model (CETP) based on knowledge graphs (KGs) to improve both global and local coherence for review generation. The proposed model learns a two-level text plan for generating a document: (1) the document plan is modeled as a sequence of sentence plans in order, and (2) the sentence plan is modeled as an entity-based subgraph from KG. Local coherence can be naturally enforced by KG subgraphs through intra-sentence correlations between entities. For global coherence, we design a hierarchical self-attentive architecture with both subgraph- and node-level attention to enhance the correlations between subgraphs. To our knowledge, we are the first to utilize a KG-based text planning model to enhance text coherence for review generation. Extensive experiments on three datasets confirm the effectiveness of our model on improving the content coherence of generated texts.

CCS CONCEPTS
• Computing methodologies → Natural language generation.

KEYWORDS
Knowledge Graph; Review Generation; Text Planning

ACM Reference Format:
Junyi Li1,2, Wayne Xin Zhao1,2, Zhicheng Wei3, Nicholas Jing Yuan3 and Ji-Rong Wen1,2. 2021. Knowledge-based Review Generation by Coherence Enhanced Text Planning. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’21).

1 INTRODUCTION
With the development of e-commerce in recent years, online reviews play a crucial role in reflecting real customer experiences. Online review information is useful to both users interested in a certain product and sellers concerned about increasing their revenue. However, many users find it tedious to write review text, and a large proportion of users do not post online reviews [4].

To ease the process of review writing, the task of review generation has been proposed and received wide attention from both research and industry communities [9, 22, 32]. Review generation aims to automatically produce review text conditioned on some necessary context inputs (e.g., users, items and ratings), which potentially influences many applications, such as explanation generation for recommendation [23], automatic scientific reviewing for papers [5]. Existing methods mainly make extensions based on sequential neural networks (e.g., recurrent neural network), including attribute awareness [9], aspect enrichment [32], and length enhancement [22]. While, these studies do not explicitly utilize the factual information about items, tending to generate dull and uninformative review text.

To enrich the generated content, we consider incorporating external knowledge graph (KG) to improve review generation. By associating KG entities with e-commerce items, we can obtain rich semantic relations about items from KG. Indeed, there has also been growing interest in the utilization of KG data in other text generation tasks, such as dialog system [33] and document summarization [2]. These approaches typically learn to copy entities or triples from KG when necessary, which improves the informativeness of the generated content to a certain extent. However, they lack overall consideration to select and arrange the incorporated KG data, which is likely to cause the issue of text incoherence [13], such as content discontinuity and logic confusion. Figure 1 presents a comparison between a coherent review and an incoherent review in terms of content continuity. As we can see, although review 2 has incorporated factual information from KG, the entire organization is poor and the review content is discontinuous in terms of semantic structure, i.e., the lead actor and actress, Leonardo Dicaprio and Kate Winslet, are separated by the genre romance film.

To address the above issue, we focus on improving entity-centric coherence of the generated reviews by leveraging the semantic structure of KGs. According to [13], entity-centric coherence refers to the entities are closely correlated to the other in text and the
entity correlations among sentences can be used to create coherence patterns for text. Also, it has been widely recognized that entity graphs (or subgraphs) are a powerful form to characterize the coherent structure in natural language [14]. Compared with the traditional entity-grid method [6] which is restricted to capturing coherent transitions between adjacent sentences, the KG-based entity graphs can easily span the entire text and capture the semantic correlations between different sentences. Following the literature of linguistics [27], we consider two kinds of coherence, namely global and local coherence. Global coherence captures how entities distribute among different sentences through inter-sentence correlations [11]. While local coherence means the intra-sentence entities (or keywords) have close correlations through semantic relations or threading words [6]. Our main idea is aimed at utilizing KG subgraphs and their correlations to enhance local and global coherence, respectively. To develop our model, we derive the generation plan before generating the text. Such a way is called text planning in text generation [17, 29], which refers to the process of selecting, arranging and ordering content to be produced.

To this end, in this paper, we propose a novel Coherence Enhanced Text Planning model (CETP) for review generation. We utilize KGs for capturing coherence patterns to generate coherent review text. Based on an augmented KG with user-item interaction and entity-keyword co-occurrence, we design a two-level text plan for generating a document: (1) the document plan is modeled as a sequence of sentence plans in order, and (2) the sentence plan is modeled as an entity-based subgraph from KG. In our model, local coherence is naturally enforced by KG subgraphs, since entities from a KG subgraph are tightly associated by semantic relations. To enhance global coherence, we develop a hierarchical self-attentive architecture with both subgraph- and node-level attention to generate a coherent sequence of subgraphs. For sentence realization, we develop a supervised copy mechanism for copying entities (or keywords) from the planned subgraph. It further improves local coherence by enhancing the intra-sentence entity correlation via threading words.

To the best of our knowledge, we are the first to utilize a KG-based text planning model to enhance both global and local coherence for review generation. For evaluation, we construct three review datasets by associating KG entities with e-commerce items to obtain semantic attributes about items. Extensive experiments demonstrate the effectiveness of our model.

2 RELATED WORK

With the striking success of deep neural networks, automatic review generation has received much attention from the research community [9, 22, 32]. Typical methods extend the Sequence-to-Sequence framework [39] by using available side information, including context information [9], sentiment score [47] and user-item interactions [31]. In order to alleviate the repetitiveness of texts caused by the RNN models, Generative Adversarial Nets (GAN) based approaches have been applied to text generation [15, 46]. Moreover, several studies utilize aspect information of products or writing style of users with a more instructive generation process to generate personalized or controllable review text [21, 22, 32]. Although various approaches have emerged, they seldom include structured attribute information about items, thus making the generated review text less informative.

In many applications [2, 21, 33], various approaches utilize structural knowledge data (e.g., Freebase and DBpedia) in the text generation process in order to improve the informativeness and diversity of the generated content. However, most of these studies mainly learn to copy related entities or triples from structural knowledge data while lack overall consideration of semantic structure of text, which have limitation in generating semantically coherent text.

Coherence is a key property of well-organized text. A variety of coherence analysis methods have been developed, such as entity grid model [6], coherence pattern model [35], and neural network model [20]. However, these methods still require considerable experience or domain expertise to define or extract features. Other related approaches include global graph model [14] which projects entities into a global graph and HMM system [26] in which the coherence between adjacent sentences is modeled by a hidden Markov framework and captured by the transition rules of topics.

Text planning is a critical component in traditional data-to-text systems [17, 29]. Typical methods are based on hand-crafted [7] or automatically-learnt rules [10], which are unable to capture rich variations of texts. Recent neural approaches mainly rely on well-designed network architectures to learn from training data [17, 37], which is difficult to control the planning process. However, as demonstrated in [43], existing neural methods are still problematic for text generation and often generate incoherent text. Moreover, these text planning methods seldom focus on review generation task, lacking consideration of user-item interaction or personalized characteristics.

3 PROBLEM FORMULATION

In this section, we introduce the notions that will be used throughout the paper, and then formally define the task.

Basic Notations. Let $\mathcal{U}$, $\mathcal{I}$ and $\mathcal{A}$ denote a user set, an item set and a rating score set, respectively. A review text is written by a user $u \in \mathcal{U}$ about an item $i \in \mathcal{I}$ with a rating score of $a \in \mathcal{A}$. Formally, a review text is denoted by $w^{1:m} = \{(w_{j,1}, \cdots, w_{j,t}, \cdots, w_{j,n_j})\}_{j=1}^m$, consisting of $m$ sentences, where $w_{j,t}$ denotes the $t$-th word (from the vocabulary $\mathcal{V}$) of the $j$-th sentence and $n_j$ is the length of the $j$-th sentence. Besides, in our setting, a knowledge graph (KG) $\mathcal{G}$ about item attributes is available for our task. Typically, it organizes facts by triples: $\mathcal{T} = \{(h, r, t)\}$, where each triple describes
that there is a relation \( r \) between head entity \( h \) and tail entity \( t \) regarding to some facts. We assume that a KG entity can be aligned to an e-commerce item. For instance, the Freebase movie entity “Avatar” (with the Freebase ID m.0bth54) has an entry of a movie item in IMDb (with the IMDb ID tt0499549). Several studies \([21, 48]\) try to develop heuristic algorithms for item-to-entity alignment and have released public linkage dataset. We add user-item links according to their interactions and entity-keyword links if they frequently co-occur in review sentences, in order to capture personalized entity preference and enhance the entity-word associations, respectively. For unifying the triple form, two kinds of non-KG links are attached with two new relations, i.e., interaction and co-occurrence. As shown in Figure 2(c), such an augmented KG can be referred as Heterogeneous KG (HKG), denoted by \( G = T \cup \{(u, r_{\text{int}}, i)\} \cup \{(e, r_{\text{co}}, w)\} \), where \( r_{\text{int}} \) and \( r_{\text{co}} \) denote the relations of user-item interaction and entity-word co-occurrence, respectively.

Planning with HKG Subgraphs. To enhance the global and local coherence, we design a two-level text plan for selecting, arranging and ordering the contents in the output text, namely document plan and sentence plan. Specifically, the document plan is modeled as a sequence of sentence plans in order, denoted by \( g^{1:m} = \langle g_1, \ldots, g_j, \ldots, g_m \rangle \). Each sentence plan is modeled as an entity-based subgraph \( g_j \) (short for subgraph) from the HKG shown in Figure 2(b), specifying the relations and entities (or keywords) to be verbalized in each sentence. We further introduce the concept of subgraph schema denoted by \( s_j \) for subgraph \( g_j \), as shown in Figure 2(a), which keeps the structure and relations of subgraphs while replaces nodes with empty slots. Typically, a subgraph schema can be instantiated into different subgraphs by filling empty slots with different entities or keywords.

Task Definition. Review generation task is concerned with how to automatically generate the review text \( w^{1:m} \) for a rating record \((u, i, a)\) based on other possible side information if any. Different from most of previous works, we incorporate the HKG \( G \) as available resource for review generation. We would like to utilize KG-based text planning model to enhance global and local coherence of the generated text.

4 The Proposed Approach

In this section, we present the proposed Coherence Enhanced Text Planning model, named CETP, for the review generation task. We first introduce the two-level text plan for selecting, arranging and ordering the contents in the output text, namely document plan and sentence plan. As discussed earlier, a document plan is modeled as a sequence of sentence plans in order and a sentence plan is modeled as an entity-based subgraph from KG. Subgraphs naturally enforce the local coherence of entities in a sentence, since they are originally connected and associated with relations in HKG. Furthermore, subgraph sequence can capture the overall distribution and arrangement of entities, which helps improve global coherence. Based on the two-level text plan, we adopt a supervised copy mechanism for sentence realization by copying entities (or keywords) from the planned subgraph. This step further improves local coherence by enhancing the intra-sentence entity correlation via threading words.

Figure 2 presents examples for illustrating the basic notations and coherence enhancement in our model.

4.1 Text Plan Generation

In this step, we study how to generate the text plan, i.e., a subgraph sequence \( g^{1:m} = \langle g_1, \ldots, g_j, \ldots, g_m \rangle \) defined in Section 3.

4.1.1 HKG Embedding. We first learn node representations for the HKG. Let \( n_j \) and \( n_k \) denote a node placeholder for the HKG, associated with an embedding vector \( v_j \in \mathbb{R}^{d_v} \), where \( d_v \) denotes the node embedding size. Node embeddings can be initialized with pre-trained KG embeddings or word embeddings \([28, 45]\). In order to capture the semantic correlations between nodes, we propose to use Relation-Enhanced Graph Transformer \([42]\), which applies a relation-enhanced multi-head attention (MHA) to obtain the node embedding \( \hat{v}_n \), for node \( n_j \) as:

\[
\hat{v}_{nj} = \text{MHA}(q_{nj}, k_{nj}, v_{nj}),
\]

where MHA(\( Q, K, V \)) follows the implementation of multi-head attention \([42]\) taking a query \( Q \), a key \( K \) and a value \( V \) as input:

\[
\text{MHA}(Q, K, V) = \text{Concat} (\text{head}_1, \ldots, \text{head}_H) W^O, \tag{2}
\]

\[
\text{head}_h = \text{Attn}(QW^Q_h, KW^K_h, VW^K_h),
\]

where \( W^Q_h, W^K_h, W^V_h \in \mathbb{R}^{d_k \times d_h} \), and \( d_h \) denotes the dimension of attention heads. For clarity, we write Eq. 1 in the form of vectors. The key point lies in that we incorporate the semantic relations between nodes into the query vector \( q_{nj} \) and key vector \( k_{nj} \):

\[
q_{nj} = v_{nj} + v_{nj} \rightarrow n_k,
\]

\[
k_{nj} = v_{nj} + v_{nk} \rightarrow n_j,
\]

where \( v_{nj} \rightarrow n_k \) and \( v_{nk} \rightarrow n_j \) are encodings for the shortest relation path \( n_j \rightarrow n_k \) and \( n_k \rightarrow n_j \) between nodes \( n_j \) and \( n_k \), which are learned by summing the embeddings of the relations in the path.

Finally, we employ a residual connection and fully connected feed-forward network (FFN):

\[
\hat{v}_{nj} = \hat{v}_{nj} + \text{FFN}(\hat{v}_{nj}),
\]

\[
\text{FFN}(x) = \max (0, x \cdot W_1 + b_1) W_2 + b_2.
\]

where \( W_1, W_2, b_1 \) and \( b_2 \) are trainable parameters, and FFN is a linear network with gelu activation. The attention block, i.e., MHA and FFN, can be stacked multiple times.

4.1.2 Subgraph Encoder. Since a sentence plan corresponds to a connected subgraph from HKG, it naturally enforces the local coherence of its nodes. The major difficulty lies in how to enhance global coherence in text planning. Let \( n_{\hat{g}_j} \in \mathbb{R}^{d_r} \) denote the embedding of the current subgraph \( g_j \), initialized by an average pooling of the embeddings of all nodes in \( g_j \) as \( n_{\hat{g}_j} = \text{AvgPooling}(\hat{v}_{nj}) \), where \( \hat{v}_{nj} \) is the embedding of node \( n_j \) in \( g_j \) learned with Graph Transformer in Section 4.1.1. Specifically, at the first step, the subgraph \( g_0 \) is initialized as a START graph without any nodes.

As shown in Figure 2(d), the nodes in previous subgraphs have closed semantic correlations with the nodes in subsequent subgraphs. Therefore, we introduce two kinds of multi-head attention to enhance global coherence for subgraph representations.
Subgraph-level Attention. To make content globally coherent, the basic idea is to refer to previous subgraphs when learning the embedding of current subgraph. For this purpose, we propose a subgraph-level multi-head attention and obtain a subgraph-enhanced subgraph embedding \( v_G^{g_j} \) as:

\[
v_G^{g_j} = \text{MHA}(v_G^{g_j}, v_{G_{j-1}}, v_{G_{j-2}}),
\]

where \( v_{G_{j-1}} \) denotes the embeddings of previous subgraphs \( g_1, \ldots, g_{j-1} \). In the subgraph-level multi-head attention mechanism, the embeddings for previous subgraphs are considered as key and value vectors, and the embedding of the current subgraph acts as the query vector. In this way, it incorporates the information of previous subgraphs for encoding the current subgraph.

Node-level Attention. Subgraph-level attention cannot directly reflect the fine-grained entity correlations between different subgraphs. Hence, we further propose to use node-level multi-head attention by considering the effect of nodes from previous subgraphs. The node-enhanced subgraph embedding \( v_N^{g_j} \) is given as:

\[
v_N^{g_j} = \text{MHA}(v_N^{g_j}, \tilde{v}_{n_2}, \tilde{v}_{n_2}),
\]

where \( \tilde{v}_{n_2} \) is the learned embedding of node \( n_2 \) in previous subgraphs \( g_1, \ldots, g_{j-1} \) computed as Eq. 4. Similar to Eq. 5, the node embeddings in previous subgraphs are considered as key and value vectors, and the embedding of the current subgraph acts as the query vector. Hence, the node information of previous subgraphs has been incorporated for encoding the current subgraph.

With the two kinds of multi-head attention, we have enhanced the global coherence by capturing inter-sentence correlations, since the information of previous subgraphs and their nodes can be injected into current subgraph. Finally, we also apply a residual connection and fully connected feed-forward network (FFN) to the node-enhanced subgraph embedding \( v_N^{g_j} \) (similar to Eq. 4) and obtain the final subgraph embedding \( \tilde{v}_{g_j} \).

4.1.3 Subgraph Decoder. After obtaining the final embedding of current subgraph \( \tilde{v}_{g_j} \), we further utilize it to generate the next subgraph, \( g_{j+1} \). General graph generation is a challenging task in deep learning [40, 41]. While, our task has several important unique characteristics, making the generation task simpler. Recall that each subgraph \( g_j \) is associated with a subgraph schema \( s_j \) and a subgraph schema can be instantiated into different subgraphs (Section 3). Thus, to generate a subgraph, we first generate the subgraph schema and then fill in the empty slots with entities or keywords. Usually, a review sentence usually contains only a few entities, and the number of frequent subgraph schemas in corpus is indeed small. We treat schema generation as a classification task over the frequent schema set, which is pre-extracted from training data. Once the schema has been determined, we utilize the relations in the schema as constraints and the entity (keyword) probability predicted by our model as selection criterion. Figure 3 presents an example for the process of subgraph generation.

To enhance the personalized characteristics of subgraphs, following Attr2Seq [9], we apply a standard attention mechanism [3] on context information \((u, i, a)\), and obtain a context vector \( \tilde{c}_j \) for encoding information of users, items and ratings. Finally, we stack the attention block, i.e., subgraph- and node-level attention, by multiple times and compute the selection probabilities for a subgraph schema and an entity (or keyword) node as:

\[
\Pr(s_{j+1}|g_1, \ldots, g_j) = \text{softmax}(W_y[\tilde{v}_{g_j}; \tilde{c}_j] + b_4),
\]

\[
\Pr(n_{j+1}|g_1, \ldots, g_j) = \text{softmax}(W_z[\tilde{v}_{g_j}; \tilde{c}_j] + b_5),
\]

where \( W_y, W_z, b_4 \) and \( b_5 \) are trainable parameters, and \( s_{j+1} \) and \( n_{j+1} \) denote a subgraph schema and an entity (or keyword), respectively. In practice, we select the most possible subgraph schema according to Eq. 7. Then, we collect all the entities that satisfy the requirement of the subgraph schema. Finally, each empty slot is filled in with the most probable node according to Eq. 8. Although there might be other combinatorial optimization strategies, our method empirically works well and is more efficient.

4.2 Sentence Realization

Given the inferred subgraph \( g_j \), we next study how to generate the words of the \( j \)-th sentence, i.e., \( \langle w_{j1}, \ldots, w_{jn}, n_{j} \rangle \).

4.2.1 Base Sentence Decoder. The base sentence generation module adopts Transformer decoder in GPT-2 [36] by stacking multiple
self-attention blocks (similar to Eq. 1–Eq. 4). Based on GPT-2, we can obtain the embedding $\hat{v}_{w_{jt}} \in \mathbb{R}^{d_w}$ for the current word $w_{jt}$ in the $j$-th sentence, where $d_w$ denotes the embedding size. Also, similar to Eq. 7–8, we follow Attr2Seq [9] to encode context information $(u, i, a)$ into a context vector $\tilde{c}_{jt}$ with attention mechanism.

We generate the next word via a softmax probability function:

$$Pr_1(w_{jt+1}|w_{j,1}, ..., w_{jt}, g_j) = \text{softmax}(W_6[\tilde{v}_{w_{jt}}, \tilde{c}_{jt}] + b_6), \quad (9)$$

where $W_6$ and $b_6$ are trainable parameters.

4.2.2 Supervised Copy Mechanism. To verbalize KG subgraphs, we introduce a supervised copy mechanism that copies nodes from the subgraph. The predictive probability of a word $w$ can be decomposed into two parts, either generating a word from the vocabulary or copying a node from the subgraph:

$$Pr(w_{jt+1} = w|w_{j,1}, ..., w_{jt}, g_j) = \lambda_{jt} \cdot Pr_1(w|w_{j,1}, ..., w_{jt}, g_j) + (1 - \lambda_{jt}) \cdot Pr_2(w|g_j),$$

where $Pr_1(w|w_{j,1}, ..., w_{jt}, g_j)$ is the generative probability from the base sentence decoder (Eq. 9), and $Pr_2(w|g_j)$ is the copy probability defined as:

$$Pr_2(w|g_j) = \frac{\exp(\text{tanh}(W_7[\tilde{v}_{w_{jt}}, \tilde{c}_{jt} ; \tilde{v}_w]))}{\sum_{w' \in g_j} \exp(\text{tanh}(W_7[\tilde{v}_{w_{jt}}, \tilde{c}_{jt} ; \tilde{v}_{w'}]))}, \quad (11)$$

where $W_7$ is the trainable parameter and $\tilde{v}_w$ is the embedding of an entity or a word node $w$ in the current subgraph $g_j$. Note that we only copy entities or keywords from the predicted subgraph, which dramatically reduces the candidate set. Since subgraph generation has already considered local and global coherence, our candidate set is more meaningful and coherent. In Eq. 10, we use a dynamically learned coefficient $\lambda_{jt}$ to control the combination between the two parts as:

$$\lambda_{jt} = \sigma(w_{\text{gen}}^T[\tilde{v}_{w_{jt}}, \tilde{c}_{jt}] + b_{\text{gen}}), \quad (12)$$

where $w_{\text{gen}}$ and $b_{\text{gen}}$ are trainable parameters. For each word, we add a binary indicator $d_{jt}$ (0 for copy and 1 for generate) to provide a supervised signal for the generation and copy. In addition to the word prediction loss, we incorporate a supervised indicator loss with the binary cross entropy:

$$L_{si} = - \sum_{j,t} d_{jt} \log(\lambda_{jt}) - (1 - d_{jt}) \log(1 - \lambda_{jt}). \quad (13)$$

Different from traditional copy mechanism, we utilize the loss in Eq. 13 to explicitly guide the switch between copy or generation during decoding, which can further enhance the local coherence via copying threading keywords from subgraphs.

4.3 Discussion and Learning

In this part, we present the model discussion and optimization.

Coherence. For local coherence, we utilize KG subgraphs as sentence plans, since KG subgraphs are tightly associated semantic structures, which naturally enforce the intra-sentence correlations of entities. Supervised copy mechanism (Section 4.2.2) further connects entities in sentences with the copied threading words from subgraphs. For global coherence, we utilize both subgraph- and node-level attention (Section 4.1.2) to enhance inter-sentence correlations of entities. Note that not all sentences include entity mentions, we set up a special sentence plan that directly calls the base decoder in Section 4.2.1 without copy mechanism. To our knowledge, there are seldom studies that consider both local and global coherence in text generation models. By incorporating KG data, our model provides a principled text planning approach for enhancing the two kinds of coherence of the generated text.

Personalization. Review generation requires to capture personalized user preference and writing styles. We explicitly model personalization through the contextual embeddings of users, items, and ratings during the decoding of subgraphs and words in Eq. 7–8 and Eq. 9–11, respectively. Another point is that HKG embedding (Section 4.1.1) has involved user-item interactions, which can capture user preference over items and associated attributes. In particular, given a (user, item) pair, we construct the HKG by involving one-hop entities linked with the item from KG and the associated keywords for entities. Such a method naturally enforces the personalized preference over item attributes for a given user.

Optimization. In our model, there are two sets of trainable parameters in subgraph generation and sentence generation, denoted by $\Theta^{(s)}$ and $\Theta^{(w)}$, respectively. First, we optimize $\Theta^{(s)}$ according to the predictive loss for subgraph schemas and nodes based on cross entropy loss using Eq. 7 and Eq. 8. And then, we optimize $\Theta^{(w)}$ according to the indicator loss in Eq. 13 and word prediction loss that sums negative likelihood of individual words using Eq. 10. We incrementally train the two parts, and fine-tune the shared or dependent parameters in different modules. For training, we directly use the real subgraphs and sentences to optimize the model parameters with Adam optimizer [18]. The same learning rate schedule in [42] is adopted in our training. In order to avoid overfitting, we adopt the dropout strategy with a ratio of 0.2. During inference, we apply our model in a pipeline way: we first infer the subgraph sequence, then predict the sentences using inferred subgraphs. For sentence generation, we apply the beam search method with a beam size of 8. We set the maximum generation lengths for subgraph and sentence sequence to be 5 and 50, respectively.

5 EXPERIMENTS

In this section, we conduct the evaluation experiments for our approach on the review generation task. We first set up the experiments, and then report the results and detailed analysis.
reinforcement learning architecture and the discriminator is a CNN-based feature extractor.

- **ACF** [22]: It decomposes the review generation process into three different stages by designing an aspect-aware coarse-to-fine generation model. The aspect semantics and syntactic characteristics are considered in the process.
- **KCGNN** [21]: It proposes a KG-enhanced review generation model based on capsule graph neural network for capturing user preference at both aspect and word levels.
- **PHVM** [38]: It adopts a planning-based hierarchical variational model to capture the inter-sentence coherence of texts.

Among these baselines, **Attr2Seq, ExpanNet, A-R2S, A2S+KG and KCGNN** are five recently proposed review generation models; **SeqGAN and LeakGAN** are GAN-based text generation models; **A2S+KG and A-R2S+KG** incorporate the pre-trained KG item embeddings from DistMult [45] and KG entities of items, respectively; **PHVM** is the state-of-the-art text planning model. We implement it by transferring KG into a list of (relation, entity) pairs (e.g., (actor, Burton)) about items (e.g., movie *Sleepy*) as inputs. We employ validation set to optimize the parameters in each method.

**Evaluation Metrics.** To evaluate the performance of review generation, we adopt two automatic generation metrics, including BLEU-1/4 and ROUGE-1/2/L. BLEU [34] measures the ratios of the co-occurrences of n-grams between the generated and real reviews; ROUGE [25] counts the overlapping n-grams between generated and real reviews. Furthermore, to evaluate the coherence of generated reviews, we adopt two automatic coherence metrics, including Sen-Sim proposed in [19] (measuring discourse coherence as an average cosine similarity between any two sentences from the discourse based on sentence embeddings from BERT [8]) and Entity Co-occurrence Ratio (abbreviated as ECR, modified based on BLEU-2 and computing the ratio of co-occurrences of entity pairs between generated and real reviews). Compared with [19] which represents the sentence embedding as the mean of embeddings of words in the sentence, BERT adds a special token “[CLS]” as the first token of every sentence and the final representation of this token is used as the sentence embedding. We also try other models (e.g., Word2Vec and ELMo) to acquire sentence embeddings. These models achieve similar results as BERT.

| Dataset                  | Electronic | Book | Movie |
|-------------------------|------------|------|-------|
| **Review**              | #Users     | 50,473 | 71,156 | 47,096 |
|                         | #Items     | 12,352 | 25,045 | 21,123 |
|                         | #Reviews   | 221,722 | 853,427 | 1,152,925 |
| **Knowledge Graph**     | #Entities  | 30,310 | 105,834 | 247,126 |
|                         | #Relations | 30     | 12     | 16     |
|                         | #Triplets  | 129,254 | 300,416 | 1,405,348 |

### 5.1 Experimental Setup

#### 5.1.1 Construction of the Datasets.

We use three datasets from different domains for evaluation, including **Amazon Electronic, Book datasets** [16], and **IMDb Movie dataset** [1]. We remove users and items occurring fewer than five times, discard reviews containing more than 100 tokens and only keep the top frequent 30,000 words in vocabulary for the three datasets. All the text is processed with the procedures of lowercase and tokenization using NLTK. In order to obtain KG information for these items, we adopt the public KB4Rec [48] dataset to construct the aligned linkage between Freebase [12] (March 2015) versions and entities. Knowledge graphs are constructed from the three domains. Starting with the aligned items as seeds, we include their one-hop neighbors and entities in the KG. We keep the reverse relations and remove the triples with non-Freebase strings. Note that we only retain the entities linked to Freebase in our datasets. The statistics of three datasets after preprocessing are summarized in Table 1. Furthermore, for each domain, we randomly split it into training, validation and test sets with a ratio of 8:1:1. To construct the entity-word links in HKG, we employ the Stanford NER package to identify entity mentions in review text, and extract aspect words by following [32]. We select 489, 442 and 440 aspect words, frequently co-occurring with entity mentions in review sentences, for the three domains, respectively. The user-item links in HKG can be constructed according to user-item interactions in review datasets. Finally, we extract the top 30, 30, and 35 frequent subgraph schemas using gSpan algorithm [44] for the three domains, respectively.

#### 5.1.2 Baseline Methods.

We consider the following baselines as comparison:

- **Attr2Seq** [9]: It adopts an attention-enhanced attribute-to-sequence framework to generate reviews with input attributes.
- **A2S+KG**: We incorporate the KG embeddings of items as additional inputs into Attr2Seq.
- **ExpanNet** [32]: It adopts an encoder-decoder architecture to generate personalized reviews by introducing aspect words.
- **A-R2S** [30]: It employs a reference-based Seq2Seq model with aspect-planning in order to cover different aspects.
- **A-R2S+KG**: We incorporate the KG entities of items as external inputs into A-R2S.
- **SeqGAN** [46]: It regards the generative model as a stochastic parameterized policy and uses Monte Carlo search to approximate the state-action value. The discriminator is a binary classifier to evaluate the sequence and guide learning process of the generator.
- **LeakGAN** [15]: It is designed for long text generation through the leaked mechanism. The generator is built upon a hierarchical aspect-planning in order to cover different aspects.

| Evaluation Metric | ExpanNet | Attr2Seq | A2S+KG | A-R2S | A-R2S+KG | SeqGAN | LeakGAN |
|-------------------|----------|----------|--------|-------|----------|--------|---------|
| BLEU-1             | 22.4     | 23.5     | 24.3   | 25.0  | 24.7     | 25.1   | 25.2    |
| ROUGE-1/L          | 48.0     | 48.5     | 48.7   | 49.0  | 49.2     | 49.3   | 49.4    |
| ECR                | 0.69     | 0.72     | 0.74   | 0.76  | 0.77     | 0.78   | 0.79    |
| Sen-Sim            | 0.86     | 0.89     | 0.91   | 0.93  | 0.94     | 0.95   | 0.95    |

Table 1: Statistics of our datasets after preprocessing.
Table 2: Performance comparisons of different methods for automatic review generation under three domains. "*" denotes the improvement is statistically significant compared with the best baseline (t-test with p-value < 0.05). "-" denotes this metric is not applicable to this method, since the generated text contains very few entity mentions.

| Datasets | Models | Coherence | Generation |
|----------|--------|-----------|------------|
|          |        | Sen-Sim   | BLEU-1 BLEU-4 ROUGE-1 ROUGE-2 ROUGE-L | |
| Electronic | Attr2Seq | 0.671 | - | 24.28 0.88 0.263 0.043 0.214 |
|          | A2S+KG | 0.652 | 1.63 | 25.62 0.93 0.271 0.049 0.223 |
|          | ExpanNet | 0.653 | - | 26.56 0.95 0.290 0.052 0.262 |
|          | A-R2S | 0.667 | - | 27.04 1.15 0.309 0.065 0.279 |
|          | A-R2S+KG | 0.650 | 7.01 | 29.28 1.69 0.322 0.067 0.288 |
|          | SeqGAN | 0.665 | - | 25.18 0.84 0.265 0.043 0.220 |
|          | LeakGAN | 0.666 | - | 25.66 0.92 0.267 0.050 0.236 |
|          | ACF | 0.686 | - | 28.22 1.04 0.315 0.066 0.280 |
|          | KCGNN | 0.688 | 16.50 | 29.88 1.83 0.323 0.078 0.295 |
|          | PHVM | 0.690 | 17.56 | 29.40 1.93 0.325 0.072 0.301 |
|          | CETP | 0.707* | 20.85* | 31.51* 3.12* 0.338* 0.084* 0.312* |
| Book | Attr2Seq | 0.677 | - | 26.93 1.14 0.259 0.047 0.223 |
|          | A2S+KG | 0.672 | 2.25 | 27.69 1.42 0.268 0.053 0.236 |
|          | ExpanNet | 0.708 | - | 26.52 1.49 0.301 0.054 0.271 |
|          | A-R2S | 0.695 | - | 28.34 1.82 0.318 0.075 0.283 |
|          | A-R2S+KG | 0.671 | 9.07 | 29.00 2.06 0.321 0.077 0.295 |
|          | SeqGAN | 0.633 | - | 26.89 1.24 0.255 0.053 0.246 |
|          | LeakGAN | 0.663 | - | 28.79 1.94 0.274 0.060 0.285 |
|          | ACF | 0.715 | - | 28.96 2.11 0.317 0.068 0.291 |
|          | KCGNN | 0.733 | 16.66 | 30.66 3.08 0.332 0.080 0.306 |
|          | PHVM | 0.740 | 18.79 | 29.33 2.46 0.319 0.085 0.307 |
|          | CETP | 0.761* | 23.89* | 31.93* 3.89* 0.341* 0.095* 0.317* |
| Movie | Attr2Seq | 0.629 | - | 26.57 1.55 0.271 0.050 0.222 |
|          | A2S+KG | 0.619 | 14.96 | 27.02 1.67 0.278 0.053 0.235 |
|          | ExpanNet | 0.651 | - | 27.93 2.00 0.301 0.063 0.266 |
|          | A-R2S | 0.649 | - | 29.01 2.12 0.314 0.074 0.306 |
|          | A-R2S+KG | 0.617 | 20.06 | 30.05 2.95 0.325 0.077 0.313 |
|          | SeqGAN | 0.641 | - | 27.07 1.63 0.274 0.052 0.221 |
|          | LeakGAN | 0.672 | - | 28.10 2.29 0.302 0.064 0.271 |
|          | ACF | 0.709 | - | 29.46 2.40 0.322 0.076 0.303 |
|          | KCGNN | 0.766 | 23.34 | 31.39 3.55 0.341 0.096 0.327 |
|          | PHVM | 0.770 | 24.70 | 30.29 3.02 0.331 0.098 0.328 |
|          | CETP | 0.794* | 28.96* | 32.37* 4.39* 0.353* 0.114* 0.345* |

and syntactic patterns, and KCGNN is a KG-enhanced generation model for capturing user preference on KG attributes. It shows that both aspect semantics and KG information are helpful for review generation, especially KG information. As the most relevant comparison with our model, the recent proposed PHVM yields the best performance among all baselines. It introduces KG entities and verbalizes coherent sentences conditioned on attribute-level planning (a sequence of entity groups).

Finally, we compare the proposed CETP with the baseline methods. It is clear to see that CETP performs better than all the baselines by a large margin. The major difference between our model and baselines lies in that we design a text planning mechanism based on KG subgraphs in the generation process, thus simultaneously improving the global and local coherence of texts. PHVM lacks the modeling of multi-grained correlations between entity groups, and also neglects the intrinsic structure of an entity group. While, other baselines do not explicitly model the coherence of text or incorporate external KG data.

5.3 Detailed Analysis
Next, we construct detailed analysis experiments on our model. We only report the results on Movie dataset due to similar findings in three datasets. We select the two best baselines KCGNN and PHVM as comparisons.

5.3.1 Ablation Analysis. Our model has three novel designs: HKG incorporation, subgraph-based text planning and supervised copy mechanism. Table 3 shows the results if we ablate these designs.
In Table 3, we can see that removing user and word nodes (including the associated links) gives a worse result than CETP, which shows that user-item interaction and entity-word co-occurrence are useful to review generation in terms of capturing user preference and entity-keyword association. Second, variants dropping the subgraph- and node-level attention are worse than CETP, especially dropping the subgraph-level attention. This shows that our model benefits from the subgraph-based text planning, which improves the process of content selection, arrangement, and order. Finally, removing the supervised copy mechanism also greatly declines the final performance of our model. In our model, the supervised copy mechanism explicitly guides the switch between generation and copy by selecting highly related entities or words from the planned subgraph, which has a significant effect on the final coherence performance.

5.3.2 Human Evaluation. Above, we have performed automatic evaluation experiments for our model and baselines. For text generation models, it is important to construct human evaluation for further effectiveness verification.

Following previous work [22, 30], we also conduct human evaluation on the generated reviews. We randomly choose 200 samples from test set. A sample contains the input contexts (i.e., user, item and rating), and the texts generated by different models. Three experienced e-commerce users were asked to score the texts with respect to four dimensions of coherence, relevance, fluency and informativeness. Coherence evaluates how content is coherent considering both intra- and inter-sentence correlation [38]. Relevance measures how relevant the generated review is according to the input contexts. Fluency measures how likely the generated review is produced by human. Informativeness means that how much the generated text provides specific or different information.

The scoring mechanism adopts a 5-point Likert scale [24], ranging from 1-point (“very terrible”) to 5-point (“very satisfying”). We further average the three scores from the three human judges over the 200 inputs for each method. The results in Table 4 show that CETP produces more coherent texts, which further verifies the effectiveness of the subgraph-based text planning. It is also worth noting that CETP performs better in terms of fluency, since KG subgraphs enforce more fluent and logical expressions. The informativeness of CETP is slightly worse than PHVM. It is possibly because PHVM applies a more greedy strategy to copy entities from KG while our model adopts a more conservative strategy to incorporate highly relevant KG entities. The Cohen’s kappa coefficients for the four factors are 0.78, 0.71, 0.75 and 0.69, respectively, indicating a high agreement between the three human judges.

5.4 Performance Sensitivity Analysis

In our paper, we have shown that KG data is very helpful to our model for both generation and coherence metrics. Here, we would examine how it affects the final performance.

5.4.1 Tuning the amount of KG data. The amount of available KG information directly affects the performance of various KG-enhanced methods. Here we examine how different methods perform with the varying amount of KG data. We select A-R2S-KG, KCGNN and PHVM as comparison methods. We take 40%, 60%, 80% and 100% of the available KG data to generate four new KG training datasets, respectively. We utilize them together with the original review data to train our model and report the performance on the test set. The KG test set is fixed as original. As shown in Figure 4(a), the performance of CETP gradually improves with the increasing amount of KG data, and CETP has achieved a consistent improvement over the other baselines with more than 40% KG data.

5.4.2 Tuning the KG embedding size. For KG data, the embedding size is an important parameter to tune in real applications, which restricts the capacity of encoding KG information. Here, we vary the embedding size in the set {64, 128, 256, 512} and construct a similar evaluation experiment as that for the amount of KG data. As we can see from Figure 4(b), CETP is substantially better than the other baselines for all the four embedding sizes, which indicates the effectiveness of our model in extracting and encoding useful

| Metrics       | Gold | CETP | KCGNN | PHVM |
|---------------|------|------|-------|------|
| Coherence     | 4.22 | 3.51 | 3.10  | 3.18 |
| Relevance     | 4.22 | 3.42 | 3.34  | 3.33 |
| Fluency       | 4.54 | 3.49 | 3.15  | 3.08 |
| Informativeness | 4.33 | 2.97 | 2.95  | 3.03 |

Figure 4: Performance tuning on Movie dataset.
Figure 5: Subgraph schema visualization and sample reviews generated by CETP on Movie dataset. The two reviews are about the movies "The Visitor" and "Moneyball" from the same user. The capital letters A, G, D, L, M and C denote the relations of actor, genre, director, language, music and co-occurrence, respectively.

5.5 Qualitative Analysis

Previous experiments have demonstrated the effectiveness of our model in generating semantically coherent review text. In this paper, we qualitatively analyze why our model performs well.

In Figure 5(a), we present the top 13 frequent subgraph schemas and their distributions from gold reviews and generated reviews of CETP and PHVM. As we can see, the distribution of subgraph schemas from CETP is closer to the real distribution (smaller MAE and RMSE results) than PHVM, indicating the effectiveness of our text planning mechanism. Furthermore, Figure 5(b) and 5(c) present two movie reviews and corresponding text plans generated by CETP for a sample user. Note that, in Figure 5(b) and 5(c), each generated sentence is verbalized from a generated subgraph (on colored background), and they are labelled with the same number. The number represent the order of the subgraph and sentence in their sequence generated by our model.

As we can see, for global coherence, CETP can capture inter-sentence entity distributions and generate similar aspect and content sketches compared with real reviews (e.g., romance film (genre)→ thomas mccarthy (director)→ richard jenkins (actor)), due to the effective text planning mechanism based on KG subgraphs. For local coherence, the sentences are well verbalized through the intra-sentence correlation between entities in subgraphs (e.g., thomas mccarthy and richard kind) and the connection of threading words (e.g., stars and performance). Besides, CETP can capture the preferred relations and entities by the user about the two movies (e.g., genre and romance film). This implies that the user-augmented KG data can provide important semantics for learning user preference.

6 CONCLUSION

In this paper, we have presented a novel coherence-enhanced text planning model for review generation. Our core idea is to utilize KG subgraphs and their correlations to enhance local and global coherence, respectively. KG subgraphs characterize the semantic structure of intra-sentence entities which can naturally enforce the local coherence since entities are tightly associated in subgraphs, while subgraph sequence can capture complicated inter-sentence correlations of entities to improve global coherence. The two kinds of coherence have been modeled in a unified, principled text planning approach based on the HKG. Furthermore, we developed a supervised copy mechanism to verbalize sentence based on KG subgraphs for further enhancing local coherence via copied threading words. We have constructed extensive experiments on three real-world review datasets. The experimental results have demonstrated the effectiveness of our model on review generation task in a series of evaluation metrics.

As future work, we will consider more kinds of external knowledge (e.g., WordNet) and investigate how our model could be applied to other domains.

ACKNOWLEDGEMENT

This work was partially supported by the National Natural Science Foundation of China under Grant No. 61872366 and 61832017, Beijing Academy of Artificial Intelligence (BAAI) under Grant No. BAAI2020J0301, Beijing Outstanding Young Scientist Program under Grant No. BJJWZYJH01201910020098, the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China under Grant No.18XNLG22 and 19XNQ047. Xin Zhao is the corresponding author.
