Set-Sequence-Graph: A Multi-View Approach Towards Exploiting Reviews for Recommendation*

Jingyue Gao
gaojingyue1997@pku.edu.cn
Peking University

Yang Lin
bdly@pku.edu.cn
Peking University

Yasha Wang†
wangyasha@pku.edu.cn
Peking University

Xiting Wang
xitwan@microsoft.com
Microsoft Research Asia

Zhao Yang
1801213718@pku.edu.cn
Peking University

Yuanduo He
andrehe@tencent.com
Tencent

Xu Chu
chu_xu@pku.edu.cn
Peking University

ABSTRACT

Existing review-based recommendation models mainly learn long-term user and item representations from a set of reviews. Due to the ignorance of rich side information of reviews, these models suffer from two drawbacks: 1) they fail to capture short-term changes of user preferences and item features reflected in reviews and 2) they cannot accurately model high-order user-item collaborative signals from reviews. To overcome these limitations, we propose a multi-view approach named **S**et-**S**equence-**G**raph (SSG), to augment existing single-view (i.e., view of set) methods by introducing two additional views of exploiting reviews: sequence and graph. In particular, with reviews organized in forms of set, sequence, and graph respectively, we design a three-way encoder architecture that jointly captures long-term (set), short-term (sequence), and collaborative (graph) features of users and items for recommendation. For the sequence encoder, we propose a short-term priority attention network that explicitly takes the order and personalized time intervals of reviews into consideration. For the graph encoder, we design a novel review-aware graph attention network to model high-order multi-aspect relations in the user-item graph. To combat the potential redundancy in captured features, our fusion module employs a cross-view decorrelation mechanism to encourage diverse representations from multiple views for integration. Experiments on public datasets demonstrate that SSG significantly outperforms state-of-the-art methods.

CCS CONCEPTS

• **Information systems → Data mining**; **Recommender systems**; • **Computing methodologies → Machine learning**.

KEYWORDS

review-based recommendation; multi-view representation learning; sequence modeling; review-aware graph attention network

ACM Reference Format:
Jingyue Gao, Yang Lin, Yasha Wang, Xiting Wang, Zhao Yang, Yuanduo He, and Xu Chu. 2020. Set-Sequence-Graph: A Multi-View Approach Towards Exploiting Reviews for Recommendation. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3340531.3411939

1 INTRODUCTION

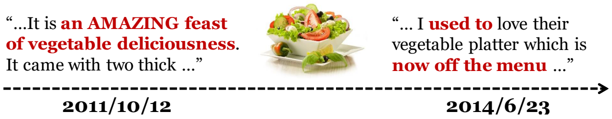
"...It is an AMAZING feast of vegetable deliciousness. It came with two thick ..."
"... I used to love their vegetable platter which is now off the menu ..."
2011/10/12
2014/6/23

Figure 1: Two reviews of a Yelp restaurant at different times. The latter indicates the change of the vegetable platter and the former is outdated. Ignoring the temporal information of reviews leads to inaccurate modeling of the restaurant.

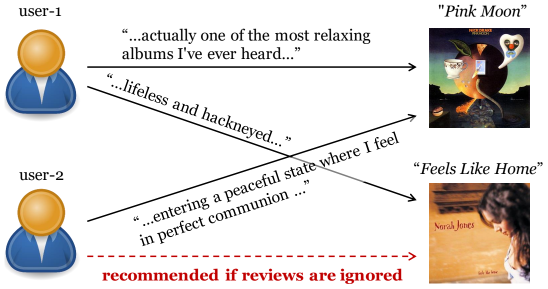
user-1
user-2
"Pink Moon"
"Feels Like Home"
"...actually one of the most relaxing albums I've ever heard..."
"...lifeless and hackneyed... "
"...entering a peaceful state where I feel in perfect communion ..."
recommended if reviews are ignored

Figure 2: A toy user-item graph of two Amazon users and albums. Considering their connectivity but ignoring reviews on edges leads to improper recommendation.

In the era of information explosion, recommender systems play an important role in helping users sift through massive choices and find suitable items. The key to accurate recommendation is properly modeling user preferences and item features based on

[*] Jingyue Gao, Yang Lin, Yasha Wang, Zhao Yang, and Xu Chu are with Key Laboratory of High Confidence Software Technologies, Ministry of Education, Beijing, China. Yasha Wang is also with National Engineering Research Center of Software Engineering, Peking University, Beijing, China.
[†] Yasha Wang is the corresponding author.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3411939

their historical interactions (e.g., ratings) [15]. Matrix Factorization (MF) [21, 33] is widely adopted to learn latent factors for users and items from the rating matrix. However, as a rating only reflects a user's overall satisfaction over an item without further details, MF methods fail to model users or items with few ratings well [42], i.e., they easily suffer from the cold-start problem.

In addition to ratings, there are abundant reviews written and shared by users on many online platforms such as Amazon and Yelp. Since reviews contain rich semantics about user preferences and item features, there has been immense research interest in exploiting them for recommendation [4, 11], where users and items are represented by their related reviews. For example, McAuley and Leskovec [31] extract latent topics from reviews and Zheng et al. [42] apply a convolutional encoder for review-based representation learning. Word-level or review-level attention is further used to highlight informative words or reviews [4, 25]. In these methods, reviews of a user/item are treated as **a set of plain texts** which are concatenated or attentively combined to capture long-term overall features. However, a review is actually more than plain text but is associated with rich side information, including its timestamp, the user who writes it, and the item which it is written for. Though existing methods have achieved encouraging performance, they suffer from two limitations due to merely focusing on the textual content of reviews and ignoring these side information.

C1: They cannot capture **short-term changes** of user preferences and item features, which often evolve instead of always being static. As shown in Fig. 1, a restaurant may update its menu every once in a while, thus leading to different user experiences as indicated in reviews. Similarly, a user's recent focus on various aspects (e.g., food, price, and service) when rating restaurants may also differ from before. Since existing methods ignore timestamps of reviews and thus fail to utilize their chronological order, they miss the opportunity to capture such short-term changes.

C2: Existing methods cannot accurately model **high-order collaborative signals** of users and items. As shown in Fig. 2, user-item interactions can form a bipartite graph where users and items are regarded as nodes, and historical interactions act as edges. In the graph, relations of nodes are revealed by high-order paths connecting them (e.g., "*Feels Like Home*"↔user-1↔"*Pink Moon*"↔ user-2), which are useful collaborative signals in recommendation since similar users tend to exhibit similar preferences on similar items. However, most review-based methods ignore this point and merely focus on the target user-item pair. Some pioneering studies model high-order node relations with graph neural networks (GNNs) on the user-item graph [40, 41]. But they ignore edge semantics carried by reviews and only consider whether two nodes are connected or not, while connectivity does not necessarily mean satisfaction or matching. For instance, the reviews in Fig. 2 show that both users enjoy "*Pink Moon*" while user-1 dislikes "*Feels Like Home*". Being unaware of the reviews on edges, existing graph methods would probably recommend "*Feels Like Home*" to user-2 as they are connected in three hops, which turns out improper.

To this end, we are motivated to incorporate these side information of reviews (i.e., the temporal information and the role in connecting users and items) to better exploit them for recommendation. Despite its necessity, there still exist several challenges. First, these side attributes of reviews are heterogeneous and accompany the same textual content in different forms (i.e., sequence-structured and graph-structured). Elaborate model design is required so that they can be properly utilized in a unified framework to provide complementary and non-redundant information for recommendation. Second, reviews are generated with irregular time intervals (shown in Fig. 3) and even the same interval could mean differently among users/items since some have reviews more frequently while others do not. Thus it is a non-trivial task to capture short-term changes with the temporal information. Third, it remains largely unexplored to incorporate edge information from natural language (i.e., reviews) into graphs to model node relations. Moreover, review semantics are complex since a user's opinion on an item may vary among different aspects (e.g., "delicious food but too expensive"), instead of simply positive or negative as indicated by the rating. Thus it is challenging to incorporate reviews into the user-item graph for accurate collaborative signals.

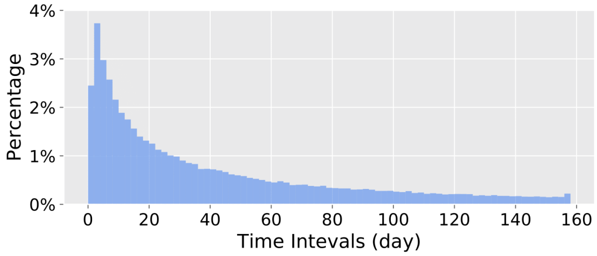

Figure 3: Distribution of time intervals between adjacent reviews in the Yelp dataset.

To tackle all these challenges, we propose a multi-view approach named **S**et-**S**equence-**G**raph (**SSG**)[1], which augments existing single-view (set) methods by introducing two additional views of exploiting reviews: sequence and graph. Particularly, we design a three-way encoder architecture for review-based representation learning. For the traditional view of set, we adopt the encoder in [4] for the long-term stable part of user preferences and item features. For the view of sequence where temporal information is introduced, we design a short-term priority encoder, which considers the order and personalized time interval of reviews for short-term representation. For the view of graph, we incorporate reviews into the user-item graph via a novel review-aware graph attention network (RGAT), which captures high-order multi-aspect relations of users and items for collaborative signals. To combat the potential redundancy caused by reuse of reviews in multiple views, we further employ a fusion module with cross-view decorrelation mechanism to encourage diversity across their representations and integrate them for final prediction.

In summary, this work makes the following key contributions:

- We propose a novel multi-view approach SSG, which employs a three-way encoder architecture and a fusion-with-decorrelation module to exploit the textual content as well as side information of reviews for recommendation. To the best of our knowledge, it is the first attempt to jointly capture long-term, short-term, and collaborative features by exploiting reviews from views of set, sequence, and graph.

[1] The source code of SSG is available at https://github.com/jygao97/SSG

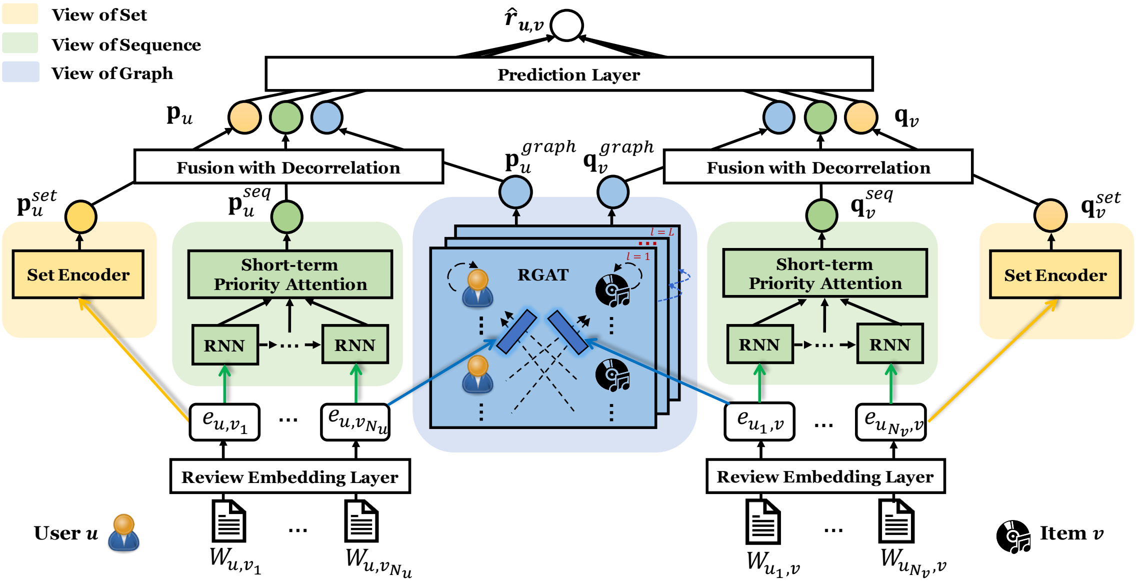

Figure 4: The framework of SSG, best viewed in color.

- We propose a short-term priority encoder for sequence modeling, which explicitly considers the order and personalized time intervals of reviews.
- We propose a novel module named RGAT to capture high-order relations of users and items, which incorporates review semantics into the process of information propagation over the user-item graph.
- We conduct extensive experiments on public datasets from different domains and demonstrate that SSG significantly outperforms state-of-the-art methods. The effectiveness of each view in SSG is also verified through ablation studies.

2 PROBLEM FORMULATION

In this section, we define our problem as follows:
Input: The input of our approach includes a user set $\mathcal{U}$, an item set $\mathcal{V}$, and a corpus of user-item interactions $\mathcal{D}$.

- Each **user** is represented by its ID $u \in \mathcal{U}$ and each **item** is represented by the item ID $v \in \mathcal{V}$.
- Each **user-item interaction** in $\mathcal{D}$ is denoted as a 5-tuple $(u, v, W_{u,v}, r_{u,v}, t_{u,v})$, where $W_{u,v}$ is the textual content of user u's review on item v, $r_{u,v}$ is the accompanying rating, and $t_{u,v}$ is the timestamp of the review. The review corpus of u is denoted by $\{W_{u,v_1}, W_{u,v_2}, ..., W_{u,v_{N_u}}\}$, where all N_u reviews of u are sorted by timestamp in ascending order. The review corpus of v is constructed similarly.

Similar to [25, 41, 42], we focus on the task of rating prediction.
Output: Given a user u and an item v, we aim to predict the rating $\hat{r}_{u,v}$ that reflects how much u likes v.

3 APPROACH

In this section, we first introduce the overview of our SSG approach. Then we detail the design of major components as well as their joint optimization, where we will focus on the part for user u since the roles of user and item are symmetric in this framework.

3.1 Overview

As shown in Fig. 4, SSG first embeds each review of the target user/item. With **review embeddings**, SSG employs a three-way encoder architecture to jointly learn user and item representations, each way corresponding to a specific view of exploiting reviews:

- **View of Set.** It captures long-term stable part of user preferences and item features. We directly adopt the model in [4] as our set encoder. It treats reviews as a set (i.e., a collection that does not specify the order of elements) of plain texts and evaluates the usefulness of each review independently. Then all reviews are combined based on their usefulness for long-term user and item representation $\mathbf{p}_u^{set}$ and $\mathbf{q}_v^{set}$.
- **View of Sequence.** It learns short-term representation $\mathbf{p}_u^{seq}$ and $\mathbf{q}_v^{seq}$ from the sequence of reviews. It first obtains the sequential representation of each review with Recurrent Neural Network (RNN). To highlight the focus on short-term features, it employs a **short-term priority attention network** to determine the relatedness of each review semantically and temporally, where the relative order and personalized time interval of reviews are also considered.
- **View of Graph.** It captures high-order collaborative features $\mathbf{p}_u^{graph}$ and $\mathbf{q}_v^{graph}$ of users and items by incorporating reviews into the user-item interaction graph. It recursively propagates node embeddings over the bipartite graph with a novel **review-aware graph attention network (RGAT)**, which consists of review-aware information propagation and multi-aspect information aggregation. In this way, review semantics can be effectively leveraged in modeling the relations among users and items.

With representations learned from three complementary views, the **fusion module** in SSG integrates them into hybrid representations for final prediction. To combat the potential redundancy caused by reuse of reviews, the fusion module is equipped with a **cross-view decorrelation** mechanism to further encourage diversity among multiple views.

3.2 Review Embedding

We calculate review embeddings with Kim CNN [17], which has shown excellent performance in capturing sentence semantics and meanwhile enables efficient computation [10, 23].

Given review $W_{u,v} = \{w_1, w_2, ..., w_n\}$ with n words, we project it into an embedding matrix $\mathbf{D} \in \mathcal{R}^{d_w \times n}$, where d_w is the word embedding size. Then a convolution operation with filter $\mathbf{f} \in \mathcal{R}^{d_w \times s}$ (s is the windows size) followed by a max pooling operation is applied to obtain a scalar feature from $\mathbf{D}$:

$$c^f = \max(\{\mathbf{f} * \mathbf{D}_{i:i+s-1}\}_{i=1}^{n-s+1}) \in \mathcal{R}, \tag{1}$$

where $*$ denotes the convolution operator. Features c^f from all d_c filters are concatenated together as the review embedding $\mathbf{e}_{u,v} = [c^{f_1}, c^{f_2}, ..., c^{f_{d_c}}] \in \mathcal{R}^{d_c}$ for $W_{u,v}$.

From section 3.3 to section 3.5, we focus on modeling u from his/her reviews. For simplicity, we drop the subscript u in $\mathbf{e}_{u,v}$ (as well as its timestamp $t_{u,v}$ and rating $r_{u,v}$) when it is unambiguous.

3.3 View of Set

Given the set $\{\mathbf{e}_{v_1}, \mathbf{e}_{v_2}, ..., \mathbf{e}_{v_N}\}$ of u's embedded reviews, we aim to learn representation that captures u's preferences over the long term. Here we adopt the encoder proposed by [4]. Based on the intuition that different reviews are not equally important in characterizing u, it attentively combines all reviews as the long-term representation $\mathbf{p}_u^{set} = \sum_{j=1}^{N} \alpha_{v_j} \mathbf{e}_{v_j}$. The the attention weight α_{v_j} of the j-th review is calculated as follows:

$$\begin{aligned} \alpha^*_{v_j} &= \mathbf{w}_1^\top \text{ReLU}(\mathbf{W}_\alpha [\mathbf{e}_{v_j}, \mathbf{i}_{v_j}] + \mathbf{b}_1) + b_2, \\ \alpha_{v_j} &= \frac{\exp(\alpha^*_{v_j})}{\sum_{j=1}^{N} \exp(\alpha^*_{v_j})}, \end{aligned} \tag{2}$$

where $\mathbf{W}_\alpha, \mathbf{w}_1, \mathbf{b}_1$ and b_2 are parameters to learn. $\mathbf{i}_{v_j}$ is the ID embedding of item v_j (i.e., the target of this review), which helps identify items whose reviews are usually more informative.

Note that the set encoder here can be replaced by many existing single-view methods. Since it is not the focus of this paper, we leave other possible choices of the set encoder for future exploration.

3.4 View of Sequence

In this view, we propose to learn short-term preferences of u from the review sequence $[\mathbf{e}_{v_1}, \mathbf{e}_{v_2}, ..., \mathbf{e}_{v_N}]$. It mainly consists of two parts: 1) the Recurrent Neural Network (RNN) and 2) the short-term priority attention network.

RNN. We first obtain sequential representations of reviews via RNN, which has been proven effective in sequence modeling. We choose Gated Recurrent Unit (GRU) [7], a variant of RNN that handles the problem of vanishing gradient. Let d_g denote the hidden size in GRU, the hidden state $\mathbf{h}_j \in \mathcal{R}^{d_g}$ of GRU is computed recursively:

$$\mathbf{h}_j = \text{GRU}(\mathbf{h}_{j-1}, \mathbf{e}_{v_j}; \Omega), \tag{3}$$

Where GRU($\cdot$) denotes the GRU unit and Ω denotes all its parameters. Thus a sequence of hidden states $\{\mathbf{h}_1, \mathbf{h}_2 ..., \mathbf{h}_N\}$ are generated, where $\mathbf{h}_j$ is the sequential representation for j-th review by characterizing the user preference up to it.

Short-term Priority Attention Network. Due to the forgetfulness and limited representation power [12] of RNN, the latest state $\mathbf{h}_N$ may not capture short-term features completely but only part of them, where previous states can serve as complements.

To this end, we propose a short-term priority attention network to determine how much a hidden state $\mathbf{h}_j$ contributes to the short-term features of $\mathbf{u}$. Particularly, with $\mathbf{h}_N$ as query, $\mathbf{h}_j$ as key and value, it learns following two score functions:

The first measures the **semantical compatibility** between $\mathbf{h}_j$ and $\mathbf{h}_N$:

$$\text{SC}(j) = \frac{(\mathbf{W}_Q \mathbf{h}_N)^\top \mathbf{W}_K \mathbf{h}_j}{\sqrt{d_g}}, \tag{4}$$

where $\mathbf{W}_Q, \mathbf{W}_K \in \mathcal{R}^{d_g \times d_g}$ are projection matrices for query and key. To avoid large values of inner product when the dimension is high, we use $\sqrt{d_g}$ as the scaling factor.

The second function measures the **temporal closeness** between $\mathbf{h}_j$ and $\mathbf{h}_N$. As they correspond to the j-th review (with timestamp t_{v_j}) and the latest review (with timestamp t_{v_N}), the relative position (i.e., order) distance of $\mathbf{h}_j$ to $\mathbf{h}_N$ is $pd_j = N - j$ and relative time interval is $ti_j = t_{v_N} - t_{v_j}$. Due to the interval irregularity in the review sequence, pd_j and ti_j can have different influence on measuring closeness and we propose to model them both:

- For pd_j, since precisely modeling it over a certain threshold may result in little gain, we clip it to $min(pd_j, z)$ and represent it with a z-dim one-hot encoding $\mathbf{pd}_j$, where the threshold z is manually specified.
- For ti_j, since some users have reviews more frequently while others are not, a personalized time interval representation is required for alignment. We first calculate the p-th percentile of time intervals $\{t_{v_j} - t_{v_{j-1}}\}_{j=2}^{N}$ between adjacent reviews of u, which is denoted as his/her base time interval ti_{base}. Then we discretize ti_j in a personalized manner similar to [24], i.e., $\lfloor \frac{ti_j}{ti_{base}} \rfloor$, followed by the operation of clip and one-hot encoding for time interval representation $\mathbf{ti}_j$.

With $\mathbf{pd}_j$ and $\mathbf{ti}_j$, we design the second score function as:

$$TC(j) = \mathbf{w}_{pd}^\top \mathbf{pd}_j + \mathbf{w}_{ti}^\top \mathbf{ti}_j, \tag{5}$$

where $\mathbf{w}_{pd}, \mathbf{w}_{ti}$ are learnable parameters. Based on the duplex score functions, the short-term priority attention layer works as:

$$\begin{aligned} \beta^*_{v_j} &= SC(j) + \lambda_{temporal} TC(j), \\ \beta_{v_j} &= \frac{\exp(\beta^*_{v_j})}{\sum_{j=1}^{N} \exp(\beta^*_{v_j})}, \\ \mathbf{p}_u^{seq} &= \sum_{j=1}^{N} \beta_{v_j} \mathbf{h}_j, \end{aligned} \tag{6}$$

where $\lambda_{temporal}$ controls the weight of two score functions. In this way, when determining the contribution of each hidden state to the short-term representation, we give priority to 1) the latest one ($\mathbf{h}_N$ is used as query) and 2) previous states which are semantically (the first score function) and temporally (the second score function) related to the latest state, which helps effectively capture short-term features from the review sequence.

3.5 View of Graph

Next, we capture high-order collaborative features by recursively propagating node embeddings over the user-item graph with a novel review-aware graph attention network (RGAT).

Review-Aware Information Propagation. Since items that u has interacted with (i.e., u's neighbors) are of different informativeness in revealing u's preferences, we exploit the idea of GAT [39] to enrich u's embedding by attentively propagating information from neighbors to u. Let $\mathbf{g}_u^{l-1}, \mathbf{g}_v^{l-1} \in \mathcal{R}^{d_{l-1}}$ denote the embedding of u and v after $l-1$ propagation layers. We first project them to a hidden space as: $\tilde{\mathbf{g}}_u^{l-1}, \tilde{\mathbf{g}}_v^{l-1} \in \mathcal{R}^{\tilde{d}_{l-1}}$ with a projection matrix $\mathbf{W}_p$. The neighborhood information of u is then calculated as:

$$\mathbf{o}_u^{l-1} = \sum_{j=1}^{N} \pi_{v_j} \tilde{\mathbf{g}}_{v_j}^{l-1}, \tag{7}$$

where π_{v_j} is the attention weight indicating the proximity of v_j to u. The vanilla GAT determines π_{v_j} merely based on the node embeddings of u and v_j. To accurately model relations between u and v_j, the first extension is to consider the rating r_{v_j}. However, r_{v_j} only indicates the overall polarity and hides detailed opinions on various aspects (e.g., a user may mention "delicious food but too expensive" in the review on a restaurant and just give a neutral rating). Thus we further incorporate rich review semantics into the attention layer:

$$\begin{aligned} \pi_{v_j}^* &= \mathbf{w}_2^\top [\tilde{\mathbf{g}}_u^{l-1}, \tilde{\mathbf{g}}_{v_j}^{l-1}, \mathbf{W}_{re}\mathbf{e}_{v_j}, \mathbf{W}_{ra}\mathbf{r}_{v_j}], \\ \pi_{v_j} &= \frac{\exp(\text{LeakyReLU}(\pi_{v_j}^*))}{\sum_{j=1}^{N} \exp(\text{LeakyReLU}(\pi_{v_j}^*))}, \end{aligned} \tag{8}$$

where LeakyReLU$(\cdot)$ is the activation function as used in the vanilla GAT, $\mathbf{W}_{re}$, $\mathbf{W}_{ra}$ and $\mathbf{w}_2$ are model parameters, and $\mathbf{r}_{v_j}$ is the one-hot encoding for the accompanying rating r_{v_j}. In this way, semantic relations between u and its neighbours are fully leveraged in the attentive information propagation process.

Multi-Aspect Information Aggregation. We derive new embedding of u by aggregating itself and its neighborhood information. We choose the sum aggregator as it balances between effectiveness and efficiency, i.e., $\mathbf{g}_u^l = \mathbf{o}_u^{l-1} + \tilde{\mathbf{g}}_u^{l-1}$.

However, u's opinions on different aspects of v_j could be various, which are hard to model with a single attention network. To capture complex semantic relations from multiple aspects, we further extend RGAT to a multi-head version. Let $\mathbf{\Theta}$ denote all parameters in the above process (i.e., the single-head version), we can reformulate the output $\mathbf{g}_u^l$ as:

$$\mathbf{g}_u^l = F(\mathbf{g}_u^{l-1}, \{\mathbf{g}_{v_j}^{l-1}, \mathbf{e}_{v_j}, r_{v_j}\}_{j=1}^N; \mathbf{\Theta}) \in \mathcal{R}^{\tilde{d}_{l-1}}. \tag{9}$$

Then output of the RGAT layer with K heads is written as:

$$\mathbf{g}_u^l = \mathop{||}_{k=1}^{K} F(\mathbf{g}_u^{l-1}, \{\mathbf{g}_{v_j}^{l-1}, \mathbf{e}_{v_j}, r_{v_j}\}_{j=1}^N; \mathbf{\Theta}_k) \in \mathcal{R}^{d_l}, \tag{10}$$

where $\mathbf{\Theta}_k$ denotes parameters of k-th head, $||$ denotes the concatenation operation, and $d_l = K\tilde{d}_{l-1}$.

By stacking L RGAT layers over the initial node embeddings (which are also treated as model parameters following [40]), we use the output $\mathbf{g}_u^L$ of the last layer as the high-order collaborative user representation $\mathbf{p}_u^{graph}$.

3.6 Fusion Module

View Integration. With $\mathbf{p}_u^{set} \in \mathcal{R}^{d_c}, \mathbf{p}_u^{seq} \in R^{d_g}$, and $\mathbf{p}_u^{graph} \in R^{d_L}$ of u captured from three views, we integrate them by feeding their concatenation into a fully-connected layer:

$$\mathbf{p}_u = \mathbf{W}_f [\mathbf{p}_u^{set}, \mathbf{p}_u^{seq}, \mathbf{p}_u^{graph}] \in \mathcal{R}^{d_f}, \tag{11}$$

where $\mathbf{W}_f$ are parameters of this layer. In this way, we obtain the hybrid user representation $\mathbf{p}_u$ that jointly captures long-term, short-term, and collaborative features. The item representation $\mathbf{q}_v$ of v are obtained similarly.

Cross-View Decorrelation. Although three views of SSG exploit reviews in different forms for different purposes, there could still exist the redundancy in their representations since they are based on the same review corpus. As redundant representations easily lead to over-fitting and bad generalizability [9], we propose to mitigate this issue via decorrelating these views.

Similar to [8, 29], we measure the correlation of two features based on their covariance. Let (x, y) denote a pair of scalar features. Their correlation can be computed as:

$$cor(x, y) = (\frac{1}{B} \sum_{b=1}^{B} (x_b - \overline{x})(y_b - \overline{y}))^2, \tag{12}$$

where B is the batch size, x_b, y_b denote their values in b-th sample, and $\overline{x}, \overline{y}$ denote their mean value in this batch. Taking the view of set and sequence as an example, the correlation of two views is thus modeled by correlations of all cross-view feature pairs:

$$cor_{set,seq} = \frac{1}{2} \sum_{i=1}^{d_c} \sum_{j=1}^{d_g} (cor(\mathbf{p}^{set}(i), \mathbf{p}^{seq}(j)) + cor(\mathbf{q}^{set}(i), \mathbf{q}^{seq}(j))), \tag{13}$$

where $\mathbf{p}^{set}(i)$ denote the i-th feature in $\mathbf{p}^{set}$. To decorrelate all three views, we impose the following loss-term:

$$\mathcal{L}_{decor} = \sum_{i,j \in \{set,seq,graph\}} cor_{i,j}. \tag{14}$$

3.7 Joint Learning

Prediction. To predict u's preference on v, we first model the feature proximity between their representations as:

$$\omega_{u,v} = (\mathbf{p}_u + \mathbf{x}_u) \odot (\mathbf{q}_v + \mathbf{y}_v), \tag{15}$$

where $\mathbf{x}_u$ and $\mathbf{y}_v$ are learnable vectors added to model their rating-related latent features that are not covered by the review-based $\mathbf{p}_u$ and $\mathbf{q}_v$. Given the interaction vector $\omega_{u,v}$, SSG predicts $\hat{r}_{u,v}$ as:

$$\hat{r}_{u,v} = \mathbf{w}_3^\top \omega_{u,v} + b_u + b_v + \mu, \tag{16}$$

where $\mathbf{w}_3$ is the weight of edges in the prediction layer. b_u, b_v, and μ are user bias, item bias, and global bias respectively.

Optimization. The objective function of SSG consists of the squared loss of prediction and the cross-view decorrelation loss:

$$\mathcal{L} = \sum_{u,v} (r_{u,v} - \hat{r}_{u,v})^2 + \lambda_{decor} \mathcal{L}_{decor}, \tag{17}$$

where $r_{u,v}$ is the ground-truth and λ_{decor} is the weight for decorrelation loss. By minimizing $\mathcal{L}$, all components in SSG can be jointly optimized in an end-to-end way.

4 EXPERIMENTS

Table 1: Statistics of four public datasets.

Dataset	#Users	#Items	#Reviews	Density
Instruments	1,429	900	10,261	0.798%
Digital Music	5,541	3,568	64,706	0.327%
Toys & Games	19,412	11,924	167,597	0.072%
Yelp	28,082	9,626	374,217	0.138%

To comprehensively evaluate our proposed SSG, we conduct experiments to answer the following research questions:

RQ1 How does SSG perform compared with state-of-the-art review-based recommendation models?
RQ2 What is the influence of each view in SSG?
RQ3 How do key hyper-parameters affect the performance of SSG, such as the dimension of representations, the number of RGAT's heads, and the weight of cross-view decorrelation?
RQ4 Is the RGAT useful in capturing collaborative signals from the view of graph that complements the view of set?

4.1 Experimental Settings

Datasets. We evaluate our approach on four public datasets with different characteristics, including three Amazon datasets[2] [14] (i.e., **Instruments**, **Digital Music**, **Toys & Games**) and the **Yelp** dataset from Yelp Challenge 2019[3] where we select restaurants located in the Phoenix city. Following [25], we use the 5-core version where all users and items have at least 5 reviews. Detailed statistics of these four datasets are summarized in Table 1.

Baselines. Eight baselines are selected for comparison, which are divided into three groups according to the type of data they use.

The first group (**G1**) consists of three methods that predict ratings only based on the observed rating matrix, including:

- **NMF** [22] applies Non-negative Matrix Factorization on the observed rating matrix to predict missing ratings.
- **PMF** [33] factorizes the rating matrix with a probabilistic linear model with Gaussian observation noise.
- **SVD++** [20] extends Singular Value Decomposition on the rating matrix with item similarities.

The second group (**G2**) exploits reviews in addition to ratings for recommendation, including:

- **HFT** [31] extracts latent topics from reviews and align them with latent factors of users and items.
- **DeepCoNN** [42] learns representations from the concatenated review document with convolutional neural networks.
- **NARRE** [4] employs the review-level attention mechanism to focus on reviews which are more useful.
- **DAML** [25] adopts local and mutual attention layers to further model the interaction of the target user and item.

The third group (**G3**) contains a graph-aware method, which considers the user-item interaction graph in addition to textual reviews:

- **RMG** [41] adopts graph neural networks to learn representations from the user-item graph for enhancement. But they fail to model reviews as edges in the graph.

Evaluation Metric. Following [25], we adopt the widely-used Mean Absolute Error (MAE) as the evaluation metric, which is calculated as:

$$MAE = \frac{1}{N}\sum_{u,v}|r_{u,v} - \hat{r}_{u,v}|, \tag{18}$$

where $r_{u,v}$ and $\hat{r}_{u,v}$ denote the actual and predicted rating respectively and N is the total number of test instances. A lower MAE indicates a better performance.

Implementation Details. Our SSG model is implemented in Pytorch[4]. We randomly split the dataset into training (80%), validation (10%), and test (10%) sets. We tune the hyper-parameters on the validation set and evaluate the performance on the test set. The hyper-parameters of baselines are reused if reported by their authors. Otherwise, we carefully tune them to ensure that they achieve the best performance. We use the Adam optimizer [18] with an initial learning rate of 0.002 and the batch size is fixed to 100. The number of latent factors d_f is tuned in [4, 8, 16, 32, 64, 128] and we set it to 8 for SSG on all datasets. For the review embedding module, we reuse the settings in the NARRE model for fair comparison, where the word embedding size is 300 and the pre-trained embedding from Google News [32] is used for initialization. The number d_c of filters in CNN is 100 and the window size is set to 3. In the view of sequence, the hidden size d_g of GRU is tuned in [4, 8, 16, 32, 64, 128]. The clip threshold z and the percentile p for the base interval are set to 100 and 10. In the view of graph, we set the number of layers and heads in RGAT to 2 and 8. λ_{decor} is tuned in [0.001, 0.01, 0.1, 1]. Each experiment is repeated ten times and we report the average and standard deviation of MAE as the result.

4.2 Overall Performance (RQ1)

The performance of our approach and baselines is shown in Table 2, from which we have the following observations.

First, methods that leverage reviews for recommendation (G2, G3, and SSG) generally perform better than those only based on ratings (G1), achieving 19.4%, 16.5%, 14.4%, and 5.4% lower MAE on average across four datasets. This is ascribed to the fact that rich semantics carried by reviews usually reveal user preferences and item features in detail and help better model users and items, while a few ratings usually fail to do so. This observation validates the necessity of exploiting reviews for recommendation.

Second, baselines that explicitly model relations of users and items (SVD++ and RMG) outperform their corresponding competitors. For example, the mean improvement of SVD++ over NMF is 13.3%. RMG also outperforms deep-learning-based methods in G2 (i.e., DeepCoNN, NARRE, and DAML) by 1.5% on average. It shows the benefit of explicitly capturing collaborative signals instead of just focusing on the target user-item pair, since users with similar preferences tend to behave similarly towards similar items. By looking at related users and items, representations of the target user/item are enriched for improvement.

[2] http://jmcauley.ucsd.edu/data/amazon
[3] https://www.yelp.com/dataset
[4] https://pytorch.org/

Table 2: Comparison among different methods. The best results are highlighted by boldface. The columns of "Impv." show the improvements of SSG over each baseline in terms of MAE. The symbol * means that the improvements over all baselines are significant with p-value < 0.01 by t-test.

Dataset		Instruments		Digital Music		Toys & Games		Yelp	
		MAE	Impv.	MAE	Impv.	MAE	Impv.	MAE	Impv.
G1	NMF	0.8399 ± 0.015	+32.0%	0.7962 ± 0.006	+23.9%	0.7546 ± 0.003	+22.1%	0.9716 ± 0.002	+7.5%
	PMF	0.8179 ± 0.015	+30.1%	0.8487 ± 0.004	+28.6%	0.7953 ± 0.004	+26.0%	0.9884 ± 0.002	+9.0%
	SVD++	0.6609 ± 0.010	+13.5%	0.6649 ± 0.003	+8.9%	0.6570 ± 0.002	+10.5%	0.9478 ± 0.001	+5.1%
G2	HFT	0.6821 ± 0.010	+16.2%	0.7194 ± 0.004	+15.8%	0.6848 ± 0.002	+14.1%	0.9504 ± 0.001	+5.4%
	DeepCoNN	0.6431 ± 0.010	+11.2%	0.6407 ± 0.005	+5.4%	0.6458 ± 0.002	+8.9%	0.9130 ± 0.001	+1.5%
	NARRE	0.6225 ± 0.011	+8.2%	0.6290 ± 0.005	+3.7%	0.6226 ± 0.002	+5.5%	0.9142 ± 0.002	+1.7%
	DAML	0.6063 ± 0.001	+5.8%	0.6407 ± 0.004	+5.4%	0.6171 ± 0.002	+4.7%	0.9178 ± 0.001	+2.0%
G3	RMG	0.6132 ± 0.011	+6.8%	0.6234 ± 0.004	+2.8%	0.6190 ± 0.002	+5.0%	0.9091 ± 0.002	+1.1%
Ours	SSG	**0.5713 ± 0.011***	-	**0.6058 ± 0.005***	-	**0.5881 ± 0.002***	-	**0.8990 ± 0.001***	-

Third, our proposed SSG achieves the best performance on all datasets, outperforming the second-best method by 5.8%, 2.8%, 4.7%, and 1.1% respectively. Its improvements over baselines are all statistically significant. This demonstrates the effectiveness of our approach, which fully exploits reviews (including textual content and valuable side information) for recommendation from three complementary views: set, sequence, and graph. The superiority of SSG mainly stems from two aspects: 1) it employs a short-term priority attention model in the view of sequence to capture short-term user preferences and item features from reviews, which are ignored by all baselines and 2) it models high-order relations between users and items while all methods except RMG fail to do so. Compared with RMG which only models the connectivity of nodes in the user-item graph, SSG achieves 3.9% lower MAE on average. We attribute it to the design that SSG further considers edge semantics revealed by reviews and thus models user-item relations more accurately.

4.3 Ablation Study (RQ2)

We analyze the influence of each view in SSG by comparing the default version with the following variants:

- *SSG without the view of set (V1)* removes the set encoder that is responsible for long-term representation.
- *SSG without the view of sequence (V2)* removes the sequence encoder that learns short-term representation.
- *SSG without the view of graph (V3)* removes the graph encoder that captures user-item collaborative signals.
- *SSG without temporal information (V4)* removes the score function of temporal closeness.
- *SSG without incorporating reviews into graph (V5)* removes reviews from the attention calculation of RGAT.

We make the following conclusions from results in Table 3.

Effectiveness of the view of set. The mean improvement of SSG over V1 on four datasets is 4.9%, which shows that the view of set is still an indispensable component in SSG. Even with short-term and collaborative features, long-term modeling of users and items plays an important role in review-based recommendation.

Table 3: Comparison among SSG and its variants. The best result on each dataset is highlighted by boldface. The symbol * means that the improvements over all variants are significant with p-value < 0.01 by t-test.

	Instruments	Digital Music	Toys & Games	Yelp
V1	0.6119 ± 0.010	0.6459 ± 0.004	0.6184 ± 0.003	0.9156 ± 0.001
V2	0.5745 ± 0.010	0.6199 ± 0.005	0.6257 ± 0.002	0.9075 ± 0.001
V3	0.6157 ± 0.011	0.6156 ± 0.005	0.6068 ± 0.002	0.9075 ± 0.002
V4	**0.5713 ± 0.011**	0.6099 ± 0.004	0.5916 ± 0.002	0.9052 ± 0.001
V5	0.6132 ± 0.009	0.6338 ± 0.004	0.6104 ± 0.002	0.9145 ± 0.001
SSG	**0.5713 ± 0.011**	**0.6058 ±0.005**	**0.5881 ± 0.002***	**0.8990 ± 0.001***

Effectiveness of the view of sequence. Compared with V2, SSG achieves 2.4% lower MAE on average. It demonstrates the effectiveness of modeling the sequence of reviews for short-term dynamic user preferences and item features. Without the sequence encoder, it would be difficult to distinguish information reflected in recent reviews from others, which results in the performance decay of V2.

Compared with V4 which discards the temporal information in the short-term priority attention layer, SSG also outperforms it on three of four datasets. It validates the necessity of explicitly considering the order and personalized time interval of reviews. We also observe that the performance gain of SSG over V2 and V4 is more significant on the last 3 datasets than Instrument. We think it is because the length and time span of sequences in the Instrument dataset are shorter than others (e.g., the average lengths of items' review sequences in the Instrument and Yelp dataset is 11 and 33, respectively), where short-term features are similar to long-term ones and make little difference to results.

Effectiveness of the view of graph. The mean improvement of SSG over V3 is 3.2% (all statistically significant with p-value < 0.01 by t-test), which validates the importance of capturing collaborative features from the user-item interaction graph. By additionally considering semantic relations of users and items, SSG enhances user and item representations and achieves better recommendation

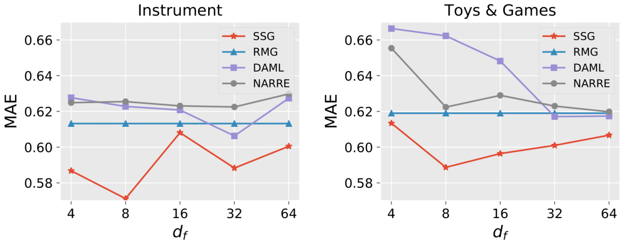

Figure 5: Performance w.r.t number of latent factors.

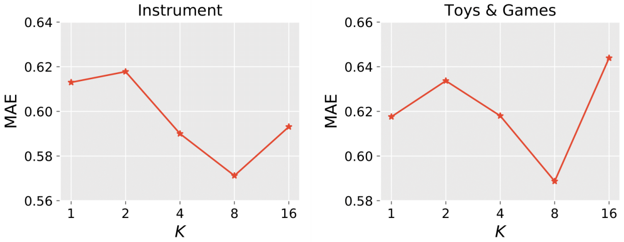

Figure 6: Performance w.r.t number of heads in RGAT.

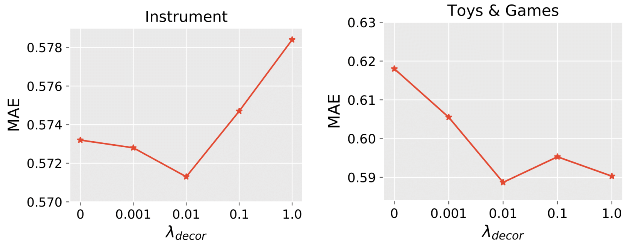

Figure 7: Performance w.r.t weight of cross-view decorrelation loss $\mathcal{L}_{decor}$.

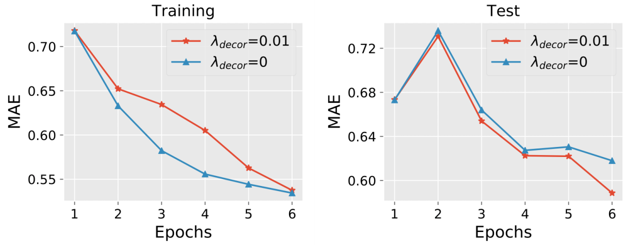

Figure 8: Performance with/without cross-view decorrelation. We show their MAE on the training (left) and test (right) set of the Toys & Games dataset during training.

performance. By contrast, V3 only utilizes reviews of the target user-item pair, which hurts the performance when there lack sufficient reviews for representation learning.

After removing reviews from the RGAT layer (V5), we further observe that the recommendation performance degrades consistently on all datasets. It verifies that merely considering the overall rating of the interaction as the edge information is not enough. It is necessary to harness review semantics to model user-item relations more accurately. Another interesting finding is that V5 sometimes even performs worse than V3 which removes the graph encoder completely. It shows that improperly modeling of the graph may introduce noisy features that impair the performance instead of bringing any benefit. All these observations show the effectiveness of our elaborately-designed RGAT for the view of graph, which achieves stable performance gain on all datasets.

4.4 Parameter Sensitivity Analysis (RQ3)

In this section, we study the effect of key hyper-parameters on model performance, including the 1) number d_f of latent factors in final user/item representation, 2) the number K of heads in RGAT, and 3) the weight λ_{decor} of the cross-view decorrelation loss $\mathcal{L}_{decor}$. Due to the space limitation, we only show sensitivity results on the Instrument dataset and Toys & Games dataset, observations on other datasets are similar.

4.4.1 Varying the number of latent factors. We vary the number of latent factors d_f in [4,8,16,32,64] and show the performance of SSG in Fig. 5. Since d_f is also a key hyper-parameter for baselines, we select the most competitive three (i.e., NARRE, DAML, and RMG) for comparison. Note that d_f is not directly adjustable in RMG, thus its performance remains the same. From Fig. 5, we find that SSG consistently outperforms baselines with varying d_f. This demonstrates the **robustness of our approach**. We also observe that SSG achieves the best performance when d_f is set to 8 and increasing the number of latent factors does not necessarily lead to improvement. We think it is because the model capacity of SSG is usually large enough due to the three-way encoder architecture, which allows it to predict accurately with concise hybrid representation.

4.4.2 Varying the number of heads in RGAT. In the view of graph, our proposed RGAT employs multiple heads to model user-item relations from multiple aspects. To study its effect, we vary the number of heads K in [1,2,4,8,16] and show the results in Fig. 6, from which we draw two conclusions. First, the performance of SSG decreases significantly when $K = 1$ (i.e., the model degenerates to a single-head version), which validates the **effectiveness of the multi-head mechanism in RGAT** since user-item relations revealed by reviews are usually too complex to be characterized with a single attention score. Second, the MAE of $K = 16$ is consistently higher than that of $K = 8$, which indicates that too many heads may cause the problem of over-fitting. We empirically find that 4 to 8 heads would be sufficient, which is also consistent with settings of researches that focus on extracting aspects from reviews [5, 6].

4.4.3 Varying the weight of cross-view decorrelation loss. In the fusion module, we impose a cross-view decorrelation loss to reduce redundancy in representations from multiple views. Here we vary its weight in [0, 0.001, 0.01, 0.1, 1.0] and show results in Fig. 7. We find that with λ_{decor} set to 0.01, SSG achieves stable performance gain compared with the setting where $\lambda_{decor} = 0$ (i.e., the cross-view decorrelation mechanism is disabled), which validates the **effectiveness of cross-view decorrelation**. We further compare the behavior of SSG with/without the decorrelation loss during the training phase until convergence, which is shown in Fig. 8: with the decorrelation loss, the training MAE decreases slower but the performance on test instances are better than the vanilla version. This observation validates our design that reducing redundancy across multiple views via decorrelation helps mitigate over-fitting and improve model generalizability.

4.5 Case Study (RQ4)

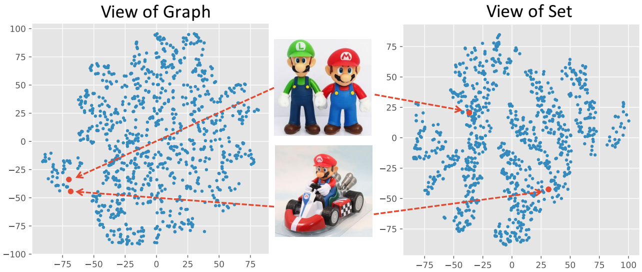

Figure 9: t-SNE visualization of representations learned by the RGAT from the view of graph (left) and by the set encoder from the view of set (right) in the Toys & Games dataset. Red points denote pairs of closely related items.

In this section, we conduct case study to investigate whether RGAT effectively captures collaborative signals so that the view of graph can complement the traditional view of set. As shown in Fig. 9, we use t-SNE [30] to visualize $\mathbf{q}_v^{graph}$ learned by the RGAT for items in the Toys & Games dataset, compared with representations learned by the single set encoder.

In the left part, data points that represent two related toy products (*Pullback Car* and *Collectible Figure Set*, which are both about the theme of *Super Mario Brothers-Nintendo*) are close to each other. It shows that our RGAT learns similar collaborative features for them based on the user-item interaction graph. After checking the related Amazon record file, we find that many users who bought the first toy also bought the second one. That is, these two toys are similar to each other from the perspective of collaborative filtering. It further verifies the reasonability of the learned representations from the view of graph. By contrast, two corresponding points in the right part are distant from each other, which means that the set encoder fails to recognize their similarity. We ascribe it to the fact that the view of set characterizes items only based on their own reviews, which is vulnerable to data insufficiency. Through the above qualitative study, we can see that the RGAT can effectively learn collaborative signals and the view of graph can well complement the existing view of set for accurate representation learning.

5 RELATED WORK

5.1 Review-based recommendation

There has been much research effort in exploiting reviews for recommendation, which mainly falls into following two categories:

Topic-Based Methods. Some works adopt topic models to extract latent topics from reviews [2, 31, 37]. For example, McAuley and Leskovec [31] propose to align latent topics extracted by Latent Dirichlet Allocation (LDA) and latent factors learned from ratings via a transform function. Bao et al. [2] derive latent topics with non-negative matrix factorization (NMF) [22] on review-word matrices. Tan et al. [37] linearly combine latent factors and latent topics to represent users and items. However, these methods organize reviews in the bag-of-words representation, which ignores the word order and fails to effectively capture semantics of reviews.

Deep-Learning-Based Methods. Recently many methods employ deep learning techniques to incorporate reviews for recommendation. For example, Zheng et al. [42] propose DeepCoNN that learns representations from the concatenated user (item) document with convolutional neural networks (CNNs) [17]. Chin et al. and Li et al. [6, 23] propose to learn aspect information with aspect-specific projection layers. Attention mechanism [1, 38] is also widely used to select informative parts of reviews to learn better representations [4, 35**?**]. For instance, [35] applied local and global attention layers to select important words from reviews. Similarly, review-level attention is designed to highlight reviews or review pairs that are more useful in rating prediction [4, 26]. More recently, Liu et al. propose DAML [25] that models interactions between user reviews and item reviews with co-attention layers. Wu et al.[41] propose RMG to enhance the representations learned from reviews with GNN on the user-item graph but it does not take reviews into consideration when constructing the graph.

All the above methods treat historical reviews of a single user (item) as a **set** of plain texts. Our proposed SSG marks a significant departure from them by introducing two novel views of organizing and exploiting reviews: **sequence** and **graph**, which enables it to additionally capture short-term and collaborative features.

5.2 Sequence Modeling with Time Intervals

Many methods for sequence modeling such as the vanilla RNN [16] implicitly assume an even distribution pattern between adjacent elements in a sequence, which often fail to handle sequences with irregular time intervals. Recently, some pioneering studies [28, 34] propose variants of RNN that are sensitive to time intervals. For example, Neil et al. [34] propose Phased LSTM that extends LSTM by adding the time gate. Baytas et al. [3] propose Time-aware LSTM that generates discounted memory according to the interval. However, these RNN methods only model time-intervals in memory flow between adjacent elements and still suffers from the problem of forgetfulness and limited representation power. In addition, Li et al. [24] extend the self-attention mechanism so that time-intervals between any two elements are modeled to capture dependencies in sequences. However, these methods are not designed to capture short-term features from sequences. Our sequence encoder differs from them in explicitly giving priority to the latest element and previous highly-related ones, where position distance and time interval to the latest element are considered in the attention layer.

5.3 Graph Neural Network

Recent years have witnessed a growing interest in modeling graph-structured data with graph neural networks (GNNs) [13, 19, 27, 36, 39]. Specifically, Kipf and Welling [19] first propose GCN that approximates a smooth filter in the spectral domain in the first order. William et al. [13] propose GraphSAGE that extends GCN from the transductive setting to inductive. To allow nodes to focus on the most relevant neighbors for aggregation, graph attention network (GAT) [39] is proposed to calculate weights among nodes with attention mechanism. In addition to homogeneous graphs, different types of edges and nodes in heterogeneous graphs are also considered by [27, 36]. For example, Shang et al. [36] model each type of edge with its own attention layer. In this paper, we consider the user-item interaction graph, where edges indicate textual reviews instead of categorical relations. Thus we propose review-aware graph attention network (RGAT) to capture complex review semantics and learn collaborative features of users/items.

6 CONCLUSION

In this paper, we propose a multi-view approach named Set-Sequence-Graph for review-based recommendation, which augments existing single-view (set) methods by introducing two additional views of exploiting reviews: sequence and graph. In this way, long-term, short-term, and collaborative features of users and items can be jointly captured. For the view of sequence, we design a short-term priority encoder that explicitly considers the order and personalized time interval of reviews. For the view of graph, we propose a novel review-aware graph attention network to model high-order relations of users and items. A fusion module is further employed to decorrelate and integrate multiple views for recommendation. Extensive experiments on public datasets validate its effectiveness.

ACKNOWLEDGMENTS

The work was supported by the National Natural Science Foundation of China (61772045) and the Project 2019BD005 supported by PKU-Baidu fund.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[2] Yang Bao, Hui Fang, and Jie Zhang. 2014. Topicmf: Simultaneously exploiting ratings and reviews for recommendation. In *AAAI*.

[3] Inci M Baytas, Cao Xiao, Xi Zhang, Fei Wang, Anil K Jain, and Jiayu Zhou. 2017. Patient subtyping via time-aware LSTM networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 65–74.

[4] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural attentional rating regression with review-level explanations. In *WWW*. 1583–1592.

[5] Zhiyong Cheng, Ying Ding, Lei Zhu, and Mohan Kankanhalli. 2018. Aspect-aware latent factor model: Rating prediction with ratings and reviews. In *Proceedings of the 2018 world wide web conference*. 639–648.

[6] Jin Yao Chin, Kaiqi Zhao, Shafiq Joty, and Gao Cong. 2018. ANR: Aspect-based neural recommender. In *CIKM*. ACM, 147–156.

[7] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).

[8] Xu Chu, Yang Lin, Yasha Wang, Leye Wang, Jiangtao Wang, and Jingyue Gao. 2019. Mlrda: A multi-task semi-supervised learning framework for drug-drug interaction prediction. In *IJCAI*. 4518–4524.

[9] Michael Cogswell, Faruk Ahmed, Ross Girshick, Larry Zitnick, and Dhruv Batra. 2016. Reducing overfitting in deep networks by decorrelating representations. In *ICLR*.

[10] Jingyue Gao, Yuanduo He, Yasha Wang, Xiting Wang, Jiangtao Wang, Guangju Peng, and Xu Chu. 2019. STAR: Spatio-Temporal Taxonomy-Aware Tag Recommendation for Citizen Complaints. In *CIKM*. 1903–1912.

[11] Jingyue Gao, Xiting Wang, Yasha Wang, and Xing Xie. 2019. Explainable recommendation through attentive multi-view learning. In *AAAI*. 3622–3629.

[12] Jingyue Gao, Xiting Wang, Yasha Wang, Zhao Yang, Junyi Gao, Jiangtao Wang, Wen Tang, and Xing Xie. 2019. Camp: Co-attention memory networks for diagnosis prediction in healthcare. In *ICDM*. 1036–1041.

[13] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in neural information processing systems*. 1024–1034.

[14] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *WWW*. 507–517.

[15] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. 173–182.

[16] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[17] Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*. 1746–1751.

[18] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[19] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[20] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *SIGKDD*. 426–434.

[21] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[22] Daniel D Lee and H Sebastian Seung. 2001. Algorithms for non-negative matrix factorization. In *NeurIPS*. 556–562.

[23] Chenliang Li, Xichuan Niu, Xiangyang Luo, Zhenzhong Chen, and Cong Quan. 2019. A Review-Driven Neural Model for Sequential Recommendation. In *IJCAI*. 2866–2872.

[24] Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time Interval Aware Self-Attention for Sequential Recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 322–330.

[25] Donghua Liu, Jing Li, Bo Du, Jun Chang, and Rong Gao. 2019. DAML: Dual Attention Mutual Learning between Ratings and Reviews for Item Recommendation. In *SIGKDD*. 344–352.

[26] Hongtao Liu, Fangzhao Wu, Wenjun Wang, Xianchen Wang, Pengfei Jiao, Chuhan Wu, and Xing Xie. 2019. NRPA: Neural Recommendation with Personalized Attention. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1233–1236.

[27] Ziqi Liu, Chaochao Chen, Xinxing Yang, Jun Zhou, Xiaolong Li, and Le Song. 2018. Heterogeneous graph neural networks for malicious account detection. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 2077–2085.

[28] Liantao Ma, Junyi Gao, Yasha Wang, Chaohe Zhang, Jiangtao Wang, Wenjie Ruan, Wen Tang, Xin Gao, and Xinyu Ma. 2020. AdaCare: Explainable Clinical Health Status Representation Learning via Scale-Adaptive Feature Extraction and Recalibration. In *AAAI*. 825–832.

[29] Liantao Ma, Chaohe Zhang, Yasha Wang, Wenjie Ruan, Jiangtao Wang, Wen Tang, Xinyu Ma, Xin Gao, and Junyi Gao. 2020. Concare: Personalized clinical feature embedding via capturing the healthcare context. In *AAAI*. 833–840.

[30] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[31] Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *RecSys*. 165–172.

[32] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NeurIPS*. 3111–3119.

[33] Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic matrix factorization. In *NeurIPS*. 1257–1264.

[34] Daniel Neil, Michael Pfeiffer, and Shih-Chii Liu. 2016. Phased lstm: Accelerating recurrent network training for long or event-based sequences. In *Advances in neural information processing systems*. 3882–3890.

[35] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *RecSys*. ACM, 297–305.

[36] Chao Shang, Qinqing Liu, Ko-Shin Chen, Jiangwen Sun, Jin Lu, Jinfeng Yi, and Jinbo Bi. 2018. Edge attention-based multi-relational graph convolutional networks. *arXiv preprint arXiv:1802.04944* (2018).

[37] Yunzhi Tan, Min Zhang, Yiqun Liu, and Shaoping Ma. 2016. Rating-boosted latent topics: Understanding users and items with ratings and reviews.. In *IJCAI*. 2640–2646.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[39] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[40] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *SIGIR*. 165–174.

[41] Chuhan Wu, Fangzhao Wu, Tao Qi, Suyu Ge, Yongfeng Huang, and Xing Xie. 2019. Reviews Meet Graphs: Enhancing User and Item Representations for Recommendation with Hierarchical Attentive Graph Neural Network. In *EMNLP-IJCNLP*. 4886–4895.

[42] Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *WSDM*. 425–434.