Topic-enhanced Graph Neural Networks for Extraction-based Explainable Recommendation

Jie Shuai
Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology
shuaijie.hfut@gmail.com

Le Wu*
Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology
lewu.ustc@gmail.com

Kun Zhang
Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology
zhang1028kun@gmail.com

Peijie Sun
Department of Computer Science and Technology, Tsinghua University
sun.hfut@gmail.com

Richang Hong
Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology
hongrc.hfut@gmail.com

Meng Wang
Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology
Hefei Comprehensive National Science Center
eric.mengwang@gmail.com

ABSTRACT

Review information has been demonstrated beneficial for the explainable recommendation. It can be treated as training corpora for generation-based methods or knowledge bases for extraction-based models. However, for generation-based methods, the sparsity of user-generated reviews and the high complexity of generative language models lead to a lack of personalization and adaptability. For extraction-based methods, focusing only on relevant attributes makes them invalid in situations where explicit attribute words are absent, limiting the potential of extraction-based models.

To this end, in this paper, we focus on the explicit and implicit analysis of review information simultaneously and propose a novel Topic-enhanced Graph Neural Networks (TGNN) to fully explore review information for better explainable recommendations. To be specific, we first use a pre-trained topic model to analyze reviews at the topic level, and design a sentence-enhanced topic graph to model user preference explicitly, where topics are intermediate nodes between users and items. Corresponding sentences serve as edge features. Thus, the requirement of explicit attribute words can be mitigated. Meanwhile, we leverage a review-enhanced rating graph to model user preference implicitly, where reviews are also considered as edge features for fine-grained user-item interaction modeling. Next, user and item representations from two graphs are used for final rating prediction and explanation extraction. Extensive experiments on three real-world datasets demonstrate the superiority of our proposed TGNN with both recommendation accuracy and explanation quality.

*Le Wu is the Corresponding author.

SIGIR '23, July 23–27, 2023, Taipei, Taiwan.

ACM ISBN 978-1-4503-9408-6/23/07…$15.00
https://doi.org/10.1145/3539618.3591776

CCS CONCEPTS

• **Information systems** → **Collaborative filtering**; **Recommender systems**.

KEYWORDS

Explainable Recommendation, Graph Neural Network, Review-based Recommendation

ACM Reference Format:
Jie Shuai, Le Wu, Kun Zhang, Peijie Sun, Richang Hong, and Meng Wang. 2023. Topic-enhanced Graph Neural Networks for Extraction-based Explainable Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3539618.3591776

1 INTRODUCTION

Explainable recommender systems not only recommend items to users but also offer corresponding explanations to depict how the recommendations are generated and why the users should pay attention to them [40, 43], so as to improve the systems' trustworthiness and persuasiveness. Apart from commonly used collaborative signals [13], additional information, such as tags [38], knowledge graph [1], and reviews [23, 27], is always employed to improve the performance of explainable recommender systems. Among them, review information is easy to collect and can provide detailed descriptions of user preferences, making it one of the most important complementary pieces of information. Plenty of work has been proposed to make full use of review information [4, 21, 22, 47].

A general idea is treating review information as corpora to train a generative model, so that explanations can be generated word by word [23, 27, 35]. However, these generation-based methods still suffer from the sparsity of available reviews, leading to the lack of personalization and adaptability of generated explanations. Moreover, Li et al. (2021a) have observed that most generated results are repetitions from the training set. Therefore, extraction-based methods are proposed to extract relevant review pieces from historical reviews [29, 40]. For example, ESCOFILT [29] leveraged K-means to

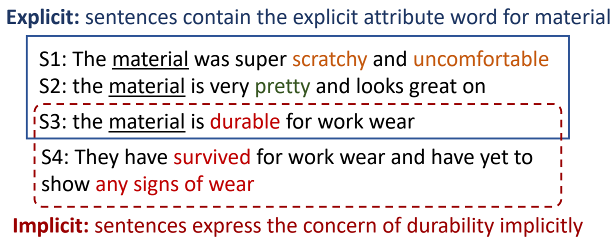

Figure 1: Reviews about material attribute explicitly (S1-S3) and implicitly (S4). S1 expresses the opinion about fitness. S2 talks about the look of the material. S3 and S4 discuss durability implicitly.

cluster similar historical reviews, so that the cluster representation can provide more comprehensive opinions to persuasive users. To further improve the personalization of explanations, EXTRA [21] was proposed to adopt traditional tensor factorization methods to measure the relevance score given a triplet of a user, an item, and an explainable sentence. GREENer [40] employed an auxiliary opinion mining toolkit [47] to extract attribute words, which were treated as intermediate to connect users and items, achieving explicit user preferences modeling on the attribute level.

Despite the achieved progress, existing extraction-based methods still suffer from some shortcomings. One of the main problems is that attribute words in reviews are not always available since users cannot always describe their opinions explicitly. Therefore, existing methods will malfunction when dealing with the review information where explicit attribute words are absent. Taking Figure 1 as an example, when focusing only on explicit attribute words, *S4* will be ignored even if it presents the same topic of interest to the user (i.e., material attribute). Moreover, focusing only on explicit attribute words cannot help to achieve comprehensive user preference modeling and convincing explanation generation, and even lead to incorrect connections between users and items. For example, although *S1*, *S2* and *S3* refer to the "material" attribute word, they discuss three different views: comfort, looks, and durability, which will import unexpected noise when taking the attribute words as intermediate to connect users and items. To this end, **how to make best use of review information is essential for extraction-based explainable recommendations**, which is also our focus in this paper.

To tackle the above challenge, we propose to take topic information into consideration. By assuming that semantically similar review sentences contain the same topic, we can extract topics from review sentences so that user preference can be measured at the topic level explicitly. Since topic information is extracted based on sentence semantics, even if user opinions are expressed implicitly in reviews, they can still be well analyzed. Therefore, the challenges turn to how to extract topic information from reviews and how to integrate topic information for personalized explanation extraction for explainable recommendations.

To this end, we propose Topic-enhanced Graph Neural Networks (TGNN) to fully exploit review information for the extraction-based explainable recommendation, in which explicit topics modeling and implicit feature learning are used for recommendation quality improvement. Specifically, for **explicit topic modeling**, inspired by the advanced topic model BERTopic [11], we adopt Infomap [30] to cluster sentence semantics with topics. Based on the topics, we devise a sentence-enhanced explicit topic graph where the topic serves as an intermediate to connect users and items. Corresponding sentences are used as edge features. Compared with existing extraction-based solutions, this graph structure more accurately models the complex relationship between the user, item, topics, and sentences. For **implicit feature learning**, we construct a review-enhanced user-item rating graph [33], where reviews also serve as edge features for better user-item interaction modeling and rating prediction. To achieve better explanation extraction, we integrate learned features from user-item rating graph with the results from the explicit topic graph, and use the integrated results to realize the sentence extraction target. Along this line, not only the rating prediction accuracy but also extracted explanation quality can be enhanced. Finally, extensive experiments over three public review datasets are conducted. The experiment results demonstrated the effectiveness and superiority of our proposed TGNN in terms of rating prediction accuracy and explanation quality. And ablation study further verified the necessity of combining the rating and topic information.

2 RELATED WORK

2.1 Review-based Explanation Models

2.1.1 Natural Language Generation-based Explanation Models. Existing review-based explanation methods mainly adopt conditional natural language generation technology to mimic real reviews word by word as explanation [27, 35, 36, 44]. However, the sparse interaction behaviors hinder the explanation model from generating diverse content in a review. Therefore, some researchers incorporate item attributes to generate diverse explanations [8, 20, 22, 27]. The item attributes are pre-extracted by a semi-supervised opinion mining toolkit, such as Sentires [15] and Snippext [26]. Instead of the semi-supervised opinion mining technique, some methods also adopt unsupervised topic models to help mine fine-grained user preferences on various topics [28, 36]. After mining explicit topic or attribute information, these methods model topic or attribute distributions in the representation of users or items and are thus used to guide the generation of diverse and personalized explanations.

2.1.2 Extraction-based Explanation models. Although generation-based models have made great progress, Li et al. observed that generation models fit the sentences in training set rather than creating new sentences. On the other hand, limited by the sparse interaction data, the generated explanations still suffer from generic content [40] and repetition issue [9]. Therefore Li et al. and Wang et al. proposed to extract human-written sentences in the training set as an explanation.

NARRE [4] is an early extraction-based explanation method that utilizes an attention mechanism to measure the usefulness score for each review. The most useful review is selected as an explanation. In addition to review-level selection, Pugoy and Kao argues the review summary could offer a better explanation and thus extractive summaries of reviews for each item as explanations. Li et al. extract sentences that co-occur across different reviews as explanations, then compare existing several ranking methods according to sentence ranking metrics. Whether language generation methods or extractive methods, recent works consider diverse topics or attributes in the review to improve explanation quality since users always express their opinions on various aspects of items [27, 36, 40]. For instance, Wang et al. extract attributions from reviews and take them as the bridge to connect user/item and sentences. However, the user-attributes-sentences graph structures may introduce noise sentences to represent users. Moreover, the attribution extraction technique [47] suffers from the domain adaption issue because of the lack of large review corpora with aspect and sentiment annotations [6, 36].

In our work, we adopt the recent advanced topic model BERTopic [11] to extract topics from reviews in an unsupervised way. After that, we introduce topics as nodes to connect users and items, where sentences are treated as fine-grained edge features to enhance user-topic and item-topic interaction modeling.

2.2 Review-based Rating Prediction Models

In a review-based recommendation system, the primary role of reviews is to extract semantic features from them to enhance user and item representations. For instance, early studies mainly adopt Latent Dirichlet Allocation [2] (LDA) to extract review topic distribution to assist user and item representation learning [25, 39]. With the remarkable advancement of deep learning in natural language processing [14, 45, 46], recent works have utilized more progressive text feature extraction methods (such as TextCNN [16], Attention [49] and BERT [7]) for review modeling and thus improving representation learning [24, 29, 48]. In addition to enhancing user and item representations, review features can be employed as regularization terms to constrain or guide the user-item interaction representation learning [3, 33, 35]. For instance, RGCL [33] takes the reviews as regularization signals to enforce the interaction representations to align to the corresponding review features at the model training stage through the contrastive learning technique.

In recent years, the neural graph networks (GNN) [18, 31] have shown an outstanding ability to model the natural user-item bipartite graph and improve recommendation performance [5, 32, 37, 41]. Therefore, several methods combine review information and user-item bipartite graph to enhance representation learning [10, 33, 42]. Among them, RMG [42] and SSG [10] both utilize Graph Attention Networks to encode the user-item graphs. However, their adopted graph attention mechanism needs to capture the complex graph patterns introduced by ratings accurately. Therefore RGCL [33] follows the idea of GC-MC, takes rating as the type of edge, and introduces comments as edge features into the graph. However, the review information is inappropriate for high-order message passing and thus can not stack multi-graph convolution layers to improve rating prediction performance.

We inherit the idea that takes review as edge features but separately model rating behavior and review information. Thus we can take advantage of high-order signals and fine-grained review information simultaneously.

3 PROBLEM DEFINITION

In review-based recommendation, there are four entity types: a user set $\mathcal{U}$ ($|\mathcal{U}| = N_u$), an item set $\mathcal{V}$ ($|\mathcal{V}| = N_v$), a rating set $\mathcal{R}$ denoting the all possible rating values (such as $\mathcal{R} = \{1, 2, 3, 4, 5\}$ in Amazon dataset) and a review set $\mathcal{E}$ representing all reviews in a dataset. An interaction record can be denoted as a quadruplet $(u_i, v_j, r_{i,j}, e_{i,j})$, which means a user $u_i \in \mathcal{U}$ give a rating score $r_{i,j} \in \mathcal{R}$ to an item $v_j \in \mathcal{V}$ with a review $e_{i,j} \in \mathcal{E}$. Moreover, a review consists of several sentences with $e_{i,j} = \{s_1, s_2, ...s_k\}$. $\mathcal{S}$ denotes the review sentence set in a dataset.

Apart from predicting the rating $\hat{r}_{i,j}$, the task of extraction-based explainable recommender system also requires an agent to retrieve several relevant sentences from sentence set $\mathcal{S}_j$ as explanations, where $\mathcal{S}_j$ is sentences of item v_j.

4 THE TECHNICAL DETAILS OF TGNN

Figure 2 illustrates the overall architecture of our proposed TGNN, including three main parts: 1) *Explicit user interesting modeling*: explicitly modeling topic information based on the newly designed Topic Graph; (2) *Implicit user interesting modeling*: implicitly modeling user interest based on the Rating Graph (3) *Topic and rating features integrating*: integrating features from two graphs for rating prediction and extraction-based explainable recommendation.

For initialization, we utilize free embeddings $\boldsymbol{U} \in \mathbb{R}^{N_u \times d}$ and $\boldsymbol{V} \in \mathbb{R}^{N_v \times d}$ to denote user and item nodes, where vectors $\boldsymbol{u}_i \in \mathbb{R}^d$ and $\boldsymbol{v}_j \in \mathbb{R}^d$ represent user u_i and item v_j respectively. With the consideration of model performance and complexity, BERT-whitening [34] is employed to encode each review and each sentence and generates corresponding feature vectors $\boldsymbol{e}_{i,j} \in \mathbb{R}^d$ and $\boldsymbol{s}_k \in \mathbb{R}^d$, respectively. Next, we will introduce each part in detail.

4.1 Explicit User Interest Modeling on Sentence-enhanced Topic Graph

In order to leverage topic information to realize the better utilization of reviews and the explicit modeling of user preference, we first construct a novel topic graph and then implement topic-level feature representation learning based on this graph.

4.1.1 Sentence-enhanced Topic Graph Construction. We construct the sentence-enhanced topic graph within two steps: topic mining and topic graph construction.

Topic Mining. Following BERTopic [11], we assume sentences that contain similar semantics have the same topic. Then, we adopt Infomap [30] to cluster sentence features encoded by BERT-Whitening [34]. Since Infomap is a clustering method for community mining in social networks, we treat sentences individually and connect two sentences according to their semantic similarity. The higher the semantic similarity between two sentences, the closer their connection. Through optimization, Infomap can automatically cluster sentences into different groups, which we regard as topics. Moreover, to control the number of topics in a relatively reasonable range, we filter out groups that contain fewer sentences as well as the corresponding sentences. Finally, we can obtain all topics and use $\mathcal{T}$ ($|\mathcal{T}| = N_t$) to represent them.

Topic Graph Construction. Each review consists of multiple sentences which are correlated to various topics. To capture user preferences on various topics, we take topics as intermediates to connect users and items, which helps align user and item representations on the topic level. One step further, considering that even

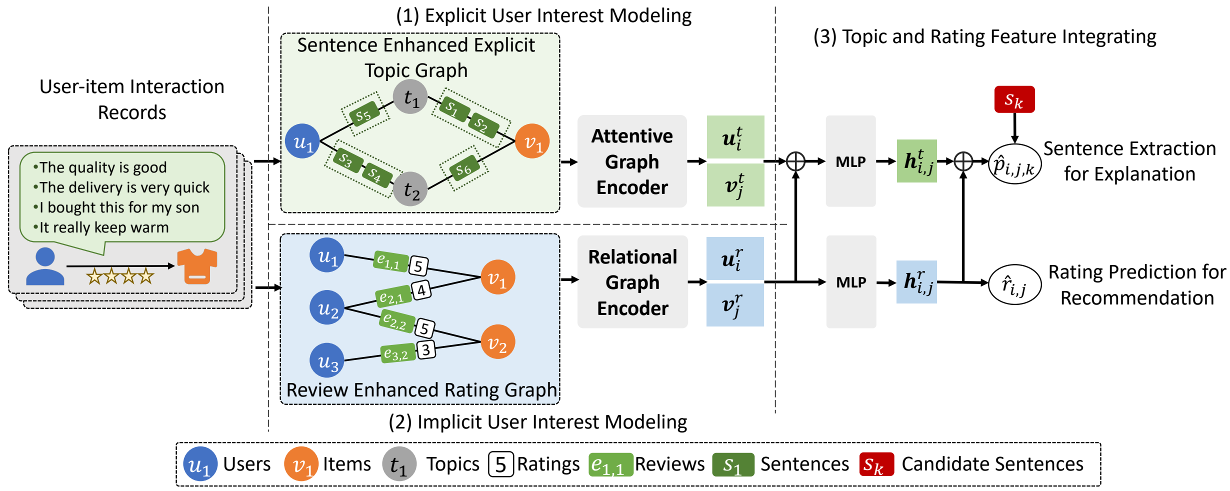

Figure 2: The architecture of Topic-enhanced Graph Neural Networks.

the same topic can have different polarities when users present their preferences, we propose to take sentences as edge features to realize the detailed analysis of topic-level use-item interactions. As shown in Figure 2 (1), if a user has mentioned a topic in his reviews, we connect the user node with this topic node. Corresponding sentences are regarded as edge features.

Specifically, we utilize matrix $\boldsymbol{C} \in \{0,1\}^{(N_u+N_v)\times N_t}$ to represent the correlation between users/items and topics. In the topic matrix $\boldsymbol{C}$, if a user or an item has mentioned a topic t, then $c_{a,t} = 1$, where we use subscript a to refer a user or an item for simple notation. Corresponding review sentences are defined as a tensor $S \in \mathcal{S}^{(N_u+N_v)\times N_t \times K}$, where each element $s_{a,t,k}$ represents the k-th sentence correlated to the topic t of user/item a. Based on the above notations, we define the topic graph as $\mathcal{G}^t =< \mathcal{U} \cup \mathcal{V} \cup \mathcal{T}, \{C, S\} >$.

4.1.2 Attentive Graph Encoder for Sentence-enhanced Topic Graph. In this part, we introduce how to learn node representation from the sentence-enhanced explicit topic graph. In short, there are two steps: 1) *topic feature learning*; 2) *topic feature aggregation*. Since similar operations are applied to learn topic-level user and item representations, we take user representation learning as an example to introduce the details.

Topic Feature Learning aims at using topic information to learn node representations. As shown in Figure 2 (1), we leverage corresponding sentences to make accurate modeling of the interaction between users and topics. Specifically, attention mechanism is employed to automatically weighted sum-up sentences to represent the explicit topic feature $\boldsymbol{\omega}_{t\to i}$ of topic t towards user u_i:

$$\boldsymbol{\omega}_{t\to i} = \boldsymbol{W} \sum_{s_k \in \mathcal{S}_{i,t}} \alpha_k^* s_k, \tag{1}$$

where $\mathcal{S}_{i,t}$ denotes the sentence set in the edge from topic t to user u_i. $\boldsymbol{s}_k \in \mathbb{R}^d$ is the k-th sentence feature vector encoded by BERT-Whitening. And $\boldsymbol{W} \in \mathbb{R}^{d\times d}$ is a trainable parameter matrix. α_k^* is the attention weights indicating the proximity of s_k to topic t.

$$\begin{aligned} \alpha_k^* &= \frac{\exp(\boldsymbol{s}_k^{*\top} \boldsymbol{o}_{i,t}^*)}{\sum_{s_k \in \mathcal{S}_{i,t}} \exp(\boldsymbol{s}_k^{*\top} \boldsymbol{o}_{i,t}^*)}, \\ \boldsymbol{o}_{i,t}^* &= \sum_{s_k \in \mathcal{S}_{i,t}} \boldsymbol{s}_k^*, \text{ where } \boldsymbol{s}_k^* = \boldsymbol{w}_t \odot \boldsymbol{s}_k, \end{aligned} \tag{2}$$

where $\boldsymbol{w}_t \in \mathbb{R}^d$ denote the t-th topic representation and $\odot$ represents Hadamard product.

Topic Feature Aggregation. After learning explicit topic features $\boldsymbol{\omega}_{t\to i}$, we next aggregate all related topic features to generate topic-level user representation. Similarly, we also leverage attention mechanism to automatically calculate the contribution of each topic and then aggregate all topic features to generate topic-level user representation u_i^t as follows:

$$\boldsymbol{u}_i^t = \mathrm{MLP}\left(\sum_{t\in\mathcal{T}_i} \beta_{t\to i}^* \boldsymbol{\omega}_{t\to i}\right), \tag{3}$$

where $\mathcal{T}_i$ represents the topic neighbor set of user u_i and $\mathrm{MLP}(\cdot)$ is a multi-layer preceptron with GELU activations. $\beta_{t\to i}^*$ is the attention weight, which is implemented as follows:

$$\beta_{t\to i}^* = \frac{\exp(\boldsymbol{\omega}_{t\to i}^\top \boldsymbol{\omega}_i^*)}{\sum_{t\in\mathcal{T}_i} \exp(\boldsymbol{\omega}_{t\to i}^\top \boldsymbol{\omega}_i^*)}, \text{ where } \boldsymbol{\omega}_i^* = \sum_{t\in\mathcal{T}_i} \boldsymbol{\omega}_{t\to i}. \tag{4}$$

Similar operations are also applied to generate topic-level item representation $\boldsymbol{v}_j^t$. Please note that we do not stack multi-layer graph convolution on the sentence-enhanced topic graph since the review information is not appropriate for high-order message passing [33].

4.2 Implicit User Interest Modeling on Review-enhanced Rating Graph

As mentioned in Section 1, due to the sparsity of user-generated reviews, it is also very essential to analyze the implicit user-item interactions (e.g., ratings) for accurate user preference modeling.

Therefore, we develop a review-enhanced rating graph [33, 37] and introduce the details in this section. Specifically, this part also consists of two steps: *Review-enhanced Rating Graph Construction* and *Relational Graph Encoder for Rating Graph.*

4.2.1 Review-enhanced Rating Graph Construction. In a review-based recommendation system, a user express his preferences for an item through a numerical rating and a textual review. The rating behavior could be denoted as a matrix $\boldsymbol{R} \in \mathcal{R}^{N_u \times N_v}$, where each element $r_{i,j} \in \mathcal{R}$ represents the rating given by user u_i to item v_j. Following RGCL [33], we define the review behavior as a matrix $\boldsymbol{E} \in \mathcal{E}^{N_u \times N_v}$, where each element $e_{i,j} \in \mathcal{E}$ is the review written by user u_i to item v_j. Based on the above rating and review behaviors, we define the rating graph as $\mathcal{G}^r =< \mathcal{U} \cup \mathcal{V}, \{\boldsymbol{R}, \boldsymbol{E}\} >$, with each edge containing a rating and a review.

4.2.2 Relational Graph Encoder for Review-enhanced Rating Graph. In order to learn implicit features accurately, we stack L graph convolution layers to capture the high-order collaborative signals. Moreover, since review information is incorporated for multi-layer message passing [33], we only introduce review information into node embeddings at the last layer. Along this line, the message passing from item v_j to user u_i at the l-th layer can be formulated as follows:

$$\boldsymbol{\mu}_{j \to i}^{(l)} = \begin{cases} \dfrac{\phi_{r_{i,j}}^{(l)}\left(\boldsymbol{e}_{i,j}\right) \boldsymbol{W}_{r_{i,j}}^{(l)} \boldsymbol{v}_j^{(l-1)}}{\sqrt{|\mathcal{N}_j^r||\mathcal{N}_i^r|}}, \text{ if } l < L \\ \dfrac{\phi_{r_{i,j}}^{(l)}\left(\boldsymbol{e}_{i,j}\right) \boldsymbol{W}_{r_{i,j}}^{(l)} \boldsymbol{v}_j^{(l-1)} + \varphi_{r_{i,j}}^{(l)}\left(\boldsymbol{e}_{i,j}\right) \mathrm{MLP}_{r_{i,j}}(\boldsymbol{e}_{i,j})}{\sqrt{|\mathcal{N}_j^r||\mathcal{N}_i^r|}}, \text{ if } l = L, \end{cases} \tag{5}$$

where $\boldsymbol{v}_j^{(l-1)}$ is item v_j's embedding learned from the $(l-1)$-th layer. We take free embeddings as the initial value of node representations. $\{\boldsymbol{W}_{\boldsymbol{r}_{i,j}}^{(l)} | r_{i,j} \in \mathcal{R}\}$ and $\{\mathrm{MLP}_{r_{i,j}} | r_{i,j} \in \mathcal{R}\}$ are trainable matrices and multi-layer perceptrons, which map node embeddings and review embeddings into the same space, respectively. Following RGCL [33], we also adopt two linear maps $\phi_{r_{i,j}}^{(l)}(\cdot)$ and $\varphi_{r_{i,j}}^{(l)}(\cdot)$ with the sigmoid function to learn two scalar weights from the review feature, which re-weight the impacts of the neighbor node and review itself on the central node. $\mathcal{N}_j^r$ and $\mathcal{N}_i^r$ represent the neighbor set of item v_j and user u_i in the rating graph $\mathcal{G}^r$. After that, we aggregate learned information from all neighbors to generate the user representation $\boldsymbol{u}_i^{(l)}$ at the l-layer. This process can be formulated as follows:

$$\boldsymbol{u}_i^{(l)} = \boldsymbol{W}^{(l)} \sum_{v_j \in \mathcal{N}_i^r} \boldsymbol{\mu}_{j \to i}^{(l)}. \tag{6}$$

After stacking L layers with GELU activation function, we transform the results from the last layer as the final implicit user representations from the rating graph:

$$\boldsymbol{u}_i^r = \boldsymbol{W} \boldsymbol{u}_i^{(L)}, \tag{7}$$

where $\boldsymbol{W}$ is the trainable parameter matrix. The implicit item representation $\boldsymbol{v}_j^r$ can be calculated analogously. Note that we use separate parameter matrices and vectors in the process of user- and item-specific side message passing and aggregation.

4.3 Integrating Topic and Rating Features For Explainable Recommendation

For extraction-based explainable recommendation problem, there are two targets: user-item rating prediction and user-item-sentence relevance score estimation.

4.3.1 Rating Prediction. With user and item representations learned from the rating graph, $\boldsymbol{u}_i^r$ and $\boldsymbol{v}_j^r$, we integrate them to obtain interaction features as follows:

$$\boldsymbol{h}_{i,j}^r = \mathrm{MLP}\left(\left[\boldsymbol{u}_i^r;\ \boldsymbol{v}_i^r\right]\right), \tag{8}$$

where [;] denotes the concatenation operation. We have tried to introduce the user and item topic representations into the rating decoder, however, either concatenation or summing up will decrease the rating prediction accuracy. We speculate the possible reason is that the topic graph focuses on modeling explicit topic information, which may disturb the original rating information. Therefore, we incorporate only node representation from the rating graph to calculate interaction features and predict final ratings. Given the interaction feature, we predict the target rating as follows:

$$\hat{r}_{i,j} = \boldsymbol{w}^\top \boldsymbol{h}_{i,j}^r, \tag{9}$$

4.3.2 Sentence Extraction. We assume a convincing recommendation explanation about a user to an item necessarily fits with her topic preferences, as well as with the received opinions of the corresponding item. Hence, we can measure the relevance score between the user, item, and sentence by the user and item topic features. Moreover, due to the sparsity problem of user-generated reviews, we incorporate implicit features from the rating graph to enhance personalized recommendation explanation extraction, which is different from the above rating prediction process. This process can be formulated as follows:

$$\boldsymbol{h}_{i,j}^t = \mathrm{MLP}\left(\left[\boldsymbol{u}_i^r + \boldsymbol{u}_i^t;\ \boldsymbol{v}_j^r + \boldsymbol{v}_j^t\right]\right). \tag{10}$$

After obtaining integrated interaction feature $\boldsymbol{h}^t$, we adopt the inner product to measure the relevance score between the interaction (u_i-v_j) and candidate sentences s_k as follows:

$$\hat{p}_{i,j,k} = \boldsymbol{s}_k^\top (\boldsymbol{h}_{i,j}^t + \boldsymbol{h}_{i,j}^r). \tag{11}$$

4.4 Model Optimization

There are two objects in our model: rating prediction and sentence retrieval. For the rating prediction task, we use Mean Square Error as the loss function:

$$\mathcal{L}_r = \frac{1}{|O|} \sum_{(i,j) \in O} (\hat{r}_{i,j} - r_{i,j})^2, \tag{12}$$

where O denotes user-item pairs in the training set and $r_{i,j}$ is the ground-truth rating. For sentence retrieval task, since our goal is to select the most relevant sentences from historical reviews, the pairwise ranking loss is used as the optimization target, which is the same as most information retrieval tasks do.

$$\mathcal{L}_s = -\frac{1}{|O|} \sum_{(i,j) \in O} \ln \sigma \left(\boldsymbol{s}_+^\top \left(\boldsymbol{h}_{i,j}^r + \boldsymbol{h}_{i,j}^t \right) - \boldsymbol{s}_-^\top \left(\boldsymbol{h}_{i,j}^r + \boldsymbol{h}_{i,j}^t \right) \right), \tag{13}$$

Table 1: Statistics of datasets.

Datasets	Clothing	CDs_and_vinyl	Yelp
#Users	39,387	75,258	52,787
#Items	23,033	64,443	21,560
#Reviews	278,677	1,097,592	633,641
Review Density	0.031%	0.023%	0.060%
#Sentences	915,972	8,384,056	3,930,864
#Topics	557	1,080	738

where $\boldsymbol{s}_+$ is the positive sentence features corresponding to the target interactions. $\boldsymbol{s}_-$ is the negative sentence features randomly sampled from the sentence set $\mathcal{S}$. We combine the rating prediction loss and sentence ranking loss as the final optimization loss:

$$\mathcal{L} = \mathcal{L}_r + \mathcal{L}_s. \tag{14}$$

5 EXPERIMENTS

5.1 Experimental Settings

5.1.1 Datasets. We conduct experiments on three datasets, including two datasets from Amazon[1] and another one from yelp[2]. The Two amazon datasets are from the "Clothing" and "CDs and Vinyl" domains. We choose the 5-core pre-processed datasets, which means that each user or item has at least five reviews. All three datasets contain user interaction records with items, including user IDs, item IDs, textual reviews, and numerical ratings. The rating values are integers from one to five. Because the raw Yelp dataset is too large, we select the records from 2019 to 2021 and keep users and items with more than ten records. The statistics of these three datasets are summarized in Table 1.

5.1.2 Implementation Details. We adopt BERT-Whitening [34] to each review or sentence and encode them into a fixed-size feature vector. The review and sentence feature vectors will not be updated during the training stage. The user and item free embeddings are initialized by the Xavier Uniform method. For simplicity, we set the size of review/sentence feature vectors and user/item embeddings as $d = 128$. We have tested the number of rating graph encoder layers from one to three. Moreover, we choose Adam [17] as the optimizer for the model training. The model is implemented with Deep Graph Library[3] and Pytorch[4]. At the test stage, given a user-item pair, we take all the sentences $\mathcal{S}_j$ of the target item v_j as candidates. According to the relevance scores inferred from Eq. (11), we select Top-N sentences the explanation.

5.2 Extractive Explanation

5.2.1 Baselines. For the extractive explanation task, we select seven baselines to compare with our model.

- **IRR**: Item Random Review is a baseline that randomly select a review from the item's historical reviews.
- **IRS**: Item Random Sentence randomly selects sentences from item's past reviews.
- **NARRE** [4]: Neural Attentional Rating Regression with Review-level Explanations is a review-level extraction method that adopts the attention scores to select reviews as explanations.
- **ESCOFILT** [29]: Extractive Summarization-based Collaborative Filtering employs cluster technique to extract summary from item past reviews as explanations.
- **CD** [21]: Canonical Decomposition is a sentence-level Tensor Factorization method, which estimates the relevance score as $\hat{s}_{i,j,k} = (\boldsymbol{u}_i \odot \boldsymbol{v}_j)^\top \boldsymbol{s}_k$.
- **PITF** [21]: Pairwise Interaction Tensor Factorization measure the relevance score as: $\hat{s}_{i,j,k} = \boldsymbol{u}_i^\top \boldsymbol{s}_k^u + \boldsymbol{v}_j^\top \boldsymbol{s}_k^v$, where $\boldsymbol{s}_k^u$ and $\boldsymbol{s}_k^v$ are user-specific and item-specific sentence embeddings. In our experiments, we take the semantic features as sentence embeddings $\boldsymbol{s}_k^u$ and $\boldsymbol{s}_k^v$.
- **GREENer** [40]: Graph-based Extractive Explainer adopts attribute words as nodes to connect users/items and sentences. We implement this model by removing the Deep Cross Network and the integer linear programming parts, because they are very time-consuming on large datasets.

5.2.2 Evaluation Metric. The explanation sentences are selected from the corresponding item's historical reviews in the training set. We take **Top-5** and **Top-10** extracted sentences as explanation results and calculate **BLEU_1**, **BLEU_2**, **BLEU_4**, **ROUGE_1_F**, **ROUGE_2_F** and **ROUGE_L_F** to automatically measure word-level overlapping between extractive explanations and the real reviews. Specifically, BLEU measures how many explanation words or segments appear in real reviews. And Rouge metric calculates how many real review words or segments appear in the extractive explanations. Following EXTRA [21], we also employ ranking-oriented metrics: Normalized Discounted Cumulative Gain (**NDCG**), Precision (**Pre**), Recall (**Rec**) and **F1** to evaluate the ranking performance of real sentences.

5.2.3 Overall Performance. Table 2 reports the extractive explanation performance comparison. According to the results, we obtain the following conclusions: First, our proposed model generally outperforms other baselines on all datasets with most metrics, which demonstrates the importance of joint modeling explicit topic and rating behavior for the extractive explanation. Our proposed sentence-enhanced topic graph accurately models the relationship between the user, item, topic, and sentences. This is the reason why our model could reach stable improvement than GREENer. Second, review-level explanation NARRE achieves better performance than IRR. This phenomenon demonstrates the effectiveness of the attention mechanism for the explanation. However, the review-level extraction limits NARRE to extracting effective explanations from the training set. And the attention mechanism is not directly optimized for the explanation purpose. Therefore, NARRE has a big performance gap with sentence-level methods. Moreover, the summarization-based method ESCOFIT takes a historical review summary as an explanation. Thus it does not optimize for the personalized explanation and performs much worse than NARRE except on ROUGE_2_F metrics. We speculate that ESCOFILT's anomalous performance on the ROUGE_2_F metric is due to the more diverse explanation provided by the summary technique.

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

Table 2: Extractive explanation performance comparison in terms of text overlapping (BLEU and ROUGE) and sentence ranking (Pre, Rec, F1, NDCG).

Methods		Text Overlapping						Sentence Ranking			
		BLEU_1	BLEU_2	BLEU_4	ROUGE_1_F	ROUGE_2_F	ROUGE_L_F	Pre	Rec	F1	NDCG
Clothing											
	IRR	15.84%	4.75%	0.99%	19.22%	1.64%	17.16%	-	-	-	-
	NARRE	18.11%	5.66%	1.05%	20.42%	1.89%	18.23%	-	-	-	-
	ESCOFIT	11.91%	4.10%	1.15%	18.56%	1.96%	16.38%	-	-	-	-
Top-5	IRS	19.06%	5.87%	1.05%	21.14%	1.96%	18.62%	0.0828	0.1271	0.0947	0.0818
	CD	18.74%	5.69%	1.03%	20.67%	1.91%	18.08%	0.0785	0.1230	0.0910	0.1929
	PITF	19.14%	6.20%	1.10%	21.70%	2.30%	18.93%	0.0599	0.1031	0.0713	0.1500
	GREENer	19.31%	6.45%	1.10%	21.94%	2.32%	18.99%	0.0933	0.1412	0.1007	0.2372
	TGNN	**19.66%****	**6.62%****	**1.18%****	**22.38%****	**2.56%****	**19.55%****	**0.1208****	**0.2137****	**0.1450****	**0.2980****
Top-10	IRS	15.11%	5.23%	0.84%	20.24%	2.20%	18.00%	0.0780	0.2213	0.1102	0.2407
	CD	**15.12%**	5.22%	0.84%	20.26%	2.21%	18.00%	0.0781	0.2208	0.1103	0.2401
	PITF	14.41%	5.32%	0.86%	20.72%	2.53%	18.43%	0.0661	0.2158	0.0969	0.2110
	GREENer	14.01%	5.27%	0.88%	20.93%	2.56%	18.52%	0.0819	0.2438	0.1172	0.2502
	TGNN	14.28%**	**5.44%****	**0.89%***	**21.08%****	**2.71%****	**18.76%****	**0.0989****	**0.3307****	**0.1458****	**0.3244****
CDs_and_Vinyl											
	IRR	15.76%	5.26%	0.84%	19.11%	2.27%	17.18%	-	-	-	-
	NARRE	16.03%	6.18%	0.92%	20.13%	2.31%	17.56%	-	-	-	-
	ESCOFIT	14.21%	4.50%	0.67%	18.60%	2.93%	15.23%	-	-	-	-
Top-5	IRS	14.58%	4.36%	0.72%	19.59%	2.06%	17.33%	0.0595	0.0431	0.0450	0.1448
	CD	16.59%	5.37%	0.93%	20.58%	2.55%	18.01%	0.0725	0.0571	0.0570	0.1713
	PITF	17.34%	6.03%	1.02%	21.65%	**3.03%**	18.93%	0.1013	0.0832	0.0803	0.2234
	GREENer	16.92%	6.07%	0.92%	21.33%	2.92%	18.91%	0.0635	0.0587	0.0581	0.1656
	TGNN	**17.76%****	**6.09%**	**1.07%****	**21.87%****	2.97%*	**19.03%****	**0.1187****	**0.1069****	**0.0985****	**0.2621****
Top-10	IRS	20.53%	7.05%	1.03%	21.13%	2.71%	18.83%	0.0623	0.0594	0.0532	0.1627
	CD	21.52%	7.78%	1.23%	21.72%	3.10%	19.20%	0.0695	0.1067	0.0761	0.2108
	PITF	21.78%	8.24%	**1.40%**	22.47%	**3.48%**	19.86%	0.0883	0.1389	0.0970	0.2565
	GREENer	21.31%	7.48 %	1.06%	21.06%	3.05%	19.21%	0.0627	0.0973	0.0742	0.2015
	TGNN	**22.34%***	**8.37%****	1.36%*	**22.83%****	**3.48%**	**20.11%****	**0.1052****	**0.1786****	**0.1187****	**0.2966****
Yelp											
	IRR	16.08%	5.36%	1.19%	18.68%	2.12%	17.13%	-	-	-	-
	NARRE	16.30%	5.42%	1.27%	18.92%	2.25%	17.29%	-	-	-	-
	ESCOFIT	14.34%	4.39%	1.02%	18.01%	2.31%	16.47%	-	-	-	-
Top-5	IRS	16.61%	4.94%	**1.01%**	19.59%	1.95%	17.73%	0.0377	0.0297	0.0305	0.0977
	CD	16.24%	4.68%	0.79%	18.79%	1.66%	16.90%	0.0538	0.0489	0.0467	0.1390
	PITF	17.62%	5.80%	0.93%	20.23%	2.13%	18.18%	0.0570	0.0540	0.0500	0.1536
	GREENer	16.54%	5.16%	0.83%	19.47%	1.96%	17.50%	0.0612	0.0487	0.4580	0.1387
	TGNN	**18.44%***	**5.89%****	1.00%	**20.54%****	**2.30%****	**18.33%****	**0.0797****	**0.0764****	**0.0705****	**0.2014****
Top-10	IRS	21.08%	7.31%	**1.35%**	21.11%	2.61%	19.23%	0.0376	0.0591	0.0425	0.1338
	CD	20.11%	6.66%	0.93%	20.41%	2.17%	18.48%	0.0477	0.0824	0.0560	0.1702
	PITF	20.32%	7.06%	1.16%	21.30%	2.65%	19.18%	0.0498	0.0908	0.0591	0.1885
	GREENer	20.88%	6.91%	1.04%	21.17%	2.56%	18.75%	0.0411	0.0685	0.0537	0.1759
	TGNN	**21.38%****	**7.52%****	1.13%**	**21.91%****	**2.74%****	**19.68%****	**0.0716****	**0.1196****	**0.0822****	**0.2364****

The best results are highlighted in bold font and the second-best results are marked by underline font. * and ** represent the statistical significance for $p < 0.05$ and $p < 0.01$, respectively, compared to the best baseline.

5.2.4 *Ablation Study.* We conduct an ablation study to further check the effectiveness of joint modeling topics and rating behaviors. We design two variants by removing the topic graph or rating graph from our model, denoted as "TGNN w/o TG" and "TGNN w/o RG" respectively. Then, we compare these two variants with our proposed TGNN on Yelp dataset in terms of text overlapping and sentence ranking metric. From Table 3, we can observe that removing either the topic graph or the rating graph will significantly reduce the explanation quality. The rating graph helps learn high-order collaborative signals, while the topic graph aims at capturing user preferences on the topic level. This ablation study verifies the importance of combining rating and topic information for recommendation explanation extraction.

5.2.5 *Topic Modeling Performance.* As mentioned in previous sections, introducing topic information from reviews are key characteristic of our proposed model, which could extract explanations from the corpus more precisely. Therefore, we continue to explore

Table 3: Ablation analysis on Yelp dataset.

Methods		Text Overlapping						Sentence Ranking			
		BLEU_1	BLEU_2	BLEU_4	ROUGE_1_F	ROUGE_2_F	ROUGE_L_F	Pre	Rec	F1	NDCG
Top-5	TGNN	**18.44%**	**5.89%**	**1.00%**	**20.54%**	**2.30%**	**18.33%**	**0.0797**	**0.0764**	**0.0705**	**0.2014**
	TGNN w/o TG	17.82%	5.71%	0.98%	20.05%	2.24%	17.96%	0.0578	0.0562	0.0514	0.1524
	TGNN w/o RG	17.89%	5.72%	0.98%	20.15%	2.26%	18.01%	0.0593	0.0569	0.0525	0.1554
Top-10	TGNN	**21.38%**	**7.52%**	**1.13%**	**21.91%**	**2.74%**	**19.68%**	**0.0716**	**0.1196**	**0.0822**	**0.2364**
	TGNN w/o TG	20.90%	7.51%	1.12%	21.41%	2.62%	19.25%	0.0528	0.0989	0.0633	0.1917
	TGNN w/o RG	21.07%	7.31%	**1.13%**	21.60%	2.74%	19.47%	0.0543	0.1007	0.0650	0.1958

Table 4: List of representative sentences for two inferred topics in Amazon Clothing dataset. The topic names are manually inferred by the corresponding sentences.

Topic	Representative Sentences
Fitness	1. They have good arch support.
	2. The arch support is very good.
	3. These have great arch support.
	4. The arch support is fantastic.
	5. They are very comfortable and have good arch support
Delivery	1. Delivery was fast and arrived before expected.
	2. It arrived three weeks before the estimated delivery date.
	3. The package was delivered even sooner than expected.
	4. It arrived quicker than expected.
	5. It arrived earlier than expect.

topic modeling to verify the effectiveness of the topic graph and better demonstrate the superiority of extractive explanations. We give a detailed analysis of topic modeling from three aspects: *the representative topic sentences*, the *topic-level accuracy of explanations*, and *Case Study* of explanation results.

The Representative Topic Sentences. To give an intuitive exhibition of the extracted topics from reviews, we list five representative sentences for two topics in the Clothing dataset in Table 4. From the result, we can observe that the representative sentences do reflect specific and meaningful topics, which are helpful for describing item attributes in different domains. For example, the sentences talking about "good arch support" or "comfortable" reflect the user preferences about the shoe item in the fitness attribution. Through these fine-grained semantics, our model could better capture the detailed user preferences and item attributes and thus can extract explanation sentences more accurately.

Topic-level accuracy of explanation. To further investigate whether the explanations extracted by our model cover topics more accurately, we measure the topic accuracy metric (Precision, Recall, and F1) between extracted explanation and the corresponding real reviews. From the evaluation result in Table 5, we can observe that our proposed model significantly outperforms baselines. This phenomenon confirms that our model reaches better text overlapping performance than baselines due to the ability to capture user preferences on topics more accurately.

Case Study. We exhibit two groups of case studies to compare the explanation result extracted by our proposed model and other baselines in Table 6. We manually highlight words that reflect the topics. From the result, we can observe that the PITF can extract explanations covering some topics that appear in the ground truth. However, it does not fare as well as our proposed model. For instance, in the first case, our model covers the "service", detailed " sandwich food and beer", and "atmosphere" topics that appear in the ground truth while PITF ignores the "good customer service" and specific "sandwich" food. Thanks to the coverage of the accurate topics, the explanation extracted by our model could better meet user preference and reach better explanation performance.

Table 5: Explanation comparison of topic-level accuracy.

Method	Top-5			Top-10		
	Pre	Rec	F1	Pre	Rec	F1
CDs_and_Vinyl						
IRS	0.1108	0.1109	0.1138	0.1083	0.1378	0.1267
CD	0.1170	0.1273	0.1215	0.1173	0.1883	0.1679
PITF	0.1235	0.1304	0.1407	0.1123	0.1976	0.1781
GREENer	0.1143	0.1273	0.1238	0.1102	0.1776	0.1635
TGNN	**0.1472**	**0.1968**	**0.1575**	**0.1294**	**0.2353**	**0.1981**
Yelp						
IRS	0.1875	0.1632	0.1753	0.1303	0.1957	0.1572
CD	0.1912	0.1980	0.1905	0.1574	0.2218	0.1609
PITF	0.1989	0.2002	0.2031	0.1671	0.2463	0.1883
GREENer	0.1892	0.1870	0.1844	0.1497	0.2038	0.1603
TGNN	**0.2062**	**0.2342**	**0.2112**	**0.1791**	**0.2641**	**0.1936**

5.3 Rating Prediction

5.3.1 Baselines. We compare our proposed TGNN with conventional CF-based models, review-based models, and state-of-the-art approaches, including (1) matrix factorization model, **SVD** [19]; (2) Neural Collaborative Filtering that captures the non-linear interaction between user and item latent factors, **NCF** [12]; (3) deep learning based solutions with reviews, **DeepCoNN** [48], **NARRE** [4], **DAML** [24], **TransNets** [3] and **ESCOFILT** [29]; (4) graph-enhanced models, **GC-MC** [37] and **RGCL** [33].

5.3.2 Overall Performance. Table 7 reports the rating prediction performance in terms of Mean Square Error on three datasets. According to the results, we can obtain the following observation: First of all, the review-based models (Table 7 (3)-(7)) achieve better performance than traditional free-embedding-based methods (Table 7 (1)-(2)), which demonstrates the effectiveness of review information for rating prediction task. Secondly, rational graph-based baselines (Table 7 (8)-(9)) model the rating behavior in the user-item bipartite graph, thus achieving better performance than other baselines. Moreover, we can notice that RGCL does not always perform better than GC-MC on all datasets, even if it incorporates review contents. This reason is the multi-layer GC-MC can leverage more higher-order neighbor information than one-layer RGCL. Therefore, we perform detailed performance comparisons according to different layers in Section 5.3.3. Third, our proposed TGNN achieves

Table 6: Example explanations extracted by several selected methods.

	Yelp
Ground Truth	... We had great **beer** the awesomest **corn appetizer, wiener, tri tip and chicken sandwich entrees** and felt they **treated us like royalty** . If you are in the mood of **good meat** and a **fun environment** .
NARRE	... The **beef** is of the highest quality ... The **Fries** are some of the best ...
PITF	They have great **beer and food** ... this place still offers **happy** hour.
GREENer	The **beer** is good and the wings are the bomb. our visit was last year ...
TGNN	The **customer service is incredible** everyone's very sweet ... the **spiked horchata and tri tip sandwich** is incredible .... Love the **atmosphere** ... All around great place to grab **good food** and **good beer**.
	Clothing
Ground Truth	These ankle boots **fit perfectly** ... I love the **thick heel** since I'm clumsy, they **give me height but are not so high that I'm wobbly** . The boots **look funky and unique**
NARRE	... the angle of the heel makes it feel higher ...
PITF	These are high quality, very attractive boots. They are so **comfortable to wear**...
GREENer	For dresses or jeans suits my style ... **easy to take arch support**. ... very **attractive boots** .
TGNN	... **had no issues with comfort**-even dancing for several hours. The **heel is not too high, just high enough** to **make them sexy** .

Table 7: Rating prediction results in terms of MSE.

Methods	Clothing	CDs_and_Vinly	Yelp
(1) SVD	1.1167	0.8662	1.1649
(2) NCF	1.1094	0.8781	1.1548
(3) DeepCoNN	1.1184	0.8621	1.1503
(4) TransNets	1.1141	0.8440	1.1491
(5) NARRE	1.1064	0.8495	1.1534
(6) ESCOFILT	1.1174	0.8633	1.1478
(7) DAML	1.1065	0.8483	1.1519
(8) GC-MC	1.0951	<u>0.8155</u>	1.1257
(9) RGCL	<u>1.0858</u>	0.8180	<u>1.1183</u>
(10) TGNN	**1.0847***	**0.8021****	**1.1132****
(11) TGNN w/o Ex	1.0913	0.8130	1.1216
(12) TGNN w/ Topic	1.1283	0.8532	1.1532

* and ** represent the statistical significance for $p < 0.05$ and $p < 0.01$, respectively, compared to the best baseline.

better performance than all baselines. TGNN takes advantage of high-order collaborative signal and fine-grained review information, thus achieving better user and item representation learning. Moreover, the ablation study Table 7(11), a variant without explanation extraction, has an apparent performance decrease, which confirms that the explanation extraction task has an enhancing effect on the rating prediction task. Another variant Table 7(12) is adding the topic interaction feature $\boldsymbol{h}_{i,j}^t$ to $\boldsymbol{h}_{i,j}^r$ for rating prediction. The large performance decrease indicates that our current explicit topic features are improper for rating prediction performance.

Table 8: Rating prediction performance comparison at different layers.

#Layers	Methods	Clothing	CDs_and_Vinly	Yelp
1 Layer	GC-MC	1.1006	0.8322	1.1259
	RGCL	<u>1.0858</u>	0.8180	<u>1.1183</u>
	TGNN	1.0868	0.8194	1.1206
2 layers	GC-MC	1.0951	<u>0.8155</u>	1.1257
	RGCL	1.0937	0.8231	1.1223
	TGNN	**1.0847***	**0.8021****	**1.1132****
3 layers	GC-MC	1.0975	0.8281	1.1445
	RGCL	1.1064	0.8296	1.1462
	TGNN	1.0925	0.8163	1.1376

5.3.3 Performance Comparison According to Different Layers. We perform a detailed comparison with GC-MC and RGCL recording to the performance at different layers (1 to 3) in Table 8. We can derive the main conclusions as follows. First, GC-MC has better performance at layer two than other layers. The two-layer GC-MC incorporate more appropriate high-order collaborative signals, thus reaching better performance. But three-layer GC-MC introduces more noisy nodes to representation learning, thus the performance slightly decreases. Second, RGCL can not capture the high-order signals and only perform best at the one-layer setting. The key reason is the review contents are inappropriate for high-order message passing in user and item representation learning. We decomposes user and item representations into high-order signals and review contents. Hence, TGNN can take both advantages of them, thus achieving the minimum rating prediction error.

6 CONCLUSION

In this paper, we proposed to exploit topic information for boosting the usage of review information, and presented a newly designed TGNN to achieve explicit and implicit analysis of review information and improve the performance of extraction-based explainable recommendations, simultaneously. Specifically, we extracted topics from reviews according to sentence semantics and then devised a sentence-enhanced topic graph, where topics serve as intermediate nodes between users and items. Therefore, user preference could be well modeled explicitly at the topic level. Meanwhile, with the consideration of the sparsity problem of user-generated reviews, we constructed a review-enhanced rating graph to implicitly model user preference. After obtaining feature representations from the sentence-enhanced topic graph and review-enhanced rating graph, we integrated them for final rating prediction and recommendation explanation extraction. Finally, extensive experiments on three large datasets demonstrated the effectiveness and the superiority of our proposed TGNN.

ACKNOWLEDGMENTS

This work is supported in part by grants from the National Key Research and Development Program of China (No. 2021ZD0111802), the National Natural Science Foundation of China (No. 72188101, No. 61972125, No. 61932009, No. U1936219, No. 62006066, No. U22A2094), Major Project of Anhui Province (No. 202203a05020011), and the fellowship of China Postdoctoral Science Foundation (No. 2022TQ0178).

REFERENCES

[1] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. 2018. Learning Heterogeneous Knowledge Base Embeddings for Explainable Recommendation. *Algorithms* (2018).

[2] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *JMLR* (2003), 993–1022.

[3] Rose Catherine and William Cohen. 2017. TransNets: Learning to Transform for Recommendation. In *RecSys*. 288–296.

[4] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural Attentional Rating Regression with Review-Level Explanations. In *WWW*. 1583–1592.

[5] Lei Chen, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. 2019. Revisiting Graph Based Collaborative Filtering: A Linear Residual Graph Convolutional Network Approach. In *AAAI*. 27–34.

[6] Hongliang Dai and Yangqiu Song. 2019. Neural Aspect and Opinion Term Extraction with Mined Rules as Weak Supervision. In *ACL*.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *EMNLP*.

[8] Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. 2017. Learning to Generate Product Reviews from Attributes. In *EACL*.

[9] Zihao Fu, Wai Lam, Anthony Man-Cho So, and Bei Shi. 2021. A Theoretical Analysis of the Repetition Problem in Text Generation.. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*.

[10] 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 *CIKM*. 395–404.

[11] Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794* (2022).

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

[13] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. 2000. Explaining Collaborative Filtering Recommendations. In *CSCW*.

[14] Zhenya Huang, Binbin Jin, Hongke Zhao, Qi Liu, Defu Lian, Tengfei Bao, and Enhong Chen. 2022. Personal or General? A Hybrid Strategy with Multi-Factors for News Recommendation. *ACM TOIS* (2022).

[15] Dongmin Hyun, Chanyoung Park, Min-Chul Yang, Ilhyeon Song, Jung-Tae Lee, and Hwanjo Yu. 2018. Review Sentiment-Guided Scalable Deep Recommender System. In *SIGIR*. 965–968.

[16] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *EMNLP*. 1746–1751.

[17] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

[18] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[19] Y. Koren, R. Bell, and C. Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.

[20] Lei Li, Yongfeng Zhang, and Li Chen. 2020. Generate Neural Template Explanations for Recommendation. In *CIKM*.

[21] Lei Li, Yongfeng Zhang, and Li Chen. 2021. EXTRA: Explanation Ranking Datasets for Explainable Recommendation. In *SIGIR*.

[22] Lei Li, Yongfeng Zhang, and Li Chen. 2021. Personalized Transformer for Explainable Recommendation. In *ACL*.

[23] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural Rating Regression with Abstractive Tips Generation for Recommendation. In *SIGIR*.

[24] 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.

[25] Julian McAuley and Jure Leskovec. 2013. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. In *RecSys*. 165–172.

[26] Zhengjie Miao, Yuliang Li, Xiaolan Wang, and Wang-Chiew Tan. 2020. Snippext: Semi-supervised Opinion Mining with Augmented Data. In *WWW*.

[27] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *EMNLP-IJCNLP*. Association for Computational Linguistics.

[28] Jianmo Ni and Julian McAuley. 2018. Personalized Review Generation By Expanding Phrases and Attending on Aspect-Aware Representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia.

[29] Reinald Adrian Pugoy and Hung-Yu Kao. 2021. Unsupervised Extractive Summarization-Based Representations for Accurate and Explainable Collaborative Filtering. In *ACL-IJCNLP*. 2981–2990.

[30] Martin Rosvall and Carl T. Bergstrom. 2008. Maps of random walks on complex networks reveal community structure. *PNAS* 105, 4 (2008), 1118–1123.

[31] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *ESWC*, Vol. 10843. 593–607.

[32] Pengyang Shao, Le Wu, Lei Chen, Kun Zhang, and Meng Wang. 2022. FairCF: Fairness-aware Collaborative Filtering. *Sci. China Inf. Sci.* (2022).

[33] Jie Shuai, Kun Zhang, Le Wu, Peijie Sun, Richang Hong, Meng Wang, and Yong Li. 2022. A Review-Aware Graph Contrastive Learning Framework for Recommendation. In *SIGIR*.

[34] Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316* (2021).

[35] Peijie Sun, Le Wu, Kun Zhang, Yanjie Fu, Richang Hong, and Meng Wang. 2020. Dual Learning for Explainable Recommendation: Towards Unifying User Preference Prediction and Review Generation. In *WWW*. 837–847.

[36] Peijie Sun, Le Wu, Kun Zhang, Yu Su, and Meng Wang. 2021. An Unsupervised Aspect-Aware Recommendation Model with Explanation Text Generation. *ACM TOIS* (2021).

[37] Rianne van den Berg, Thomas N. Kipf, and Max Welling. 2017. Graph Convolutional Matrix Completion. *KDD Deep Learning Day* (2017).

[38] Jesse Vig, Shilad Sen, and John Riedl. 2009. Tagsplanations: Explaining Recommendations Using Tags. In *IUI*.

[39] Chong Wang and David M. Blei. 2011. Collaborative Topic Modeling for Recommending Scientific Articles. In *SIGKDD*. 448–456.

[40] Peng Wang, Renqin Cai, and Hongning Wang. 2022. Graph-based Extractive Explainer for Recommendations. In *WWW*.

[41] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Clicks can be Cheating: Counterfactual Recommendation for Mitigating Clickbait Issue. In *SIGIR*. 1288–1297.

[42] 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*. 4884–4893.

[43] Le Wu, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang. 2022. A Survey on Accuracy-oriented Neural Recommendation: From Collaborative Filtering to Information-rich Recommendation. *TKDE* (2022).

[44] Aobo Yang, Nan Wang, Renqin Cai, Hongbo Deng, and Hongning Wang. 2022. Comparative Explanations of Recommendations. In *Proceedings of the ACM Web Conference 2022*. 3113–3123. https://doi.org/10.1145/3485447.3512031

[45] Kun Zhang, Guangyi Lv, Le Wu, Enhong Chen, Qi Liu, and Meng Wang. 2021. LadRa-Net: Locally Aware Dynamic Reread Attention Net for Sentence Semantic Matching. *IEEE TNNLS* (2021).

[46] Kun Zhang, Le Wu, Guangyi Lv, Meng Wang, Enhong Chen, and Shulan Ruan. 2021. Making the Relation Matters: Relation of Relation Learning Network for Sentence Semantic Matching.

[47] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *SIGIR*.

[48] Lei Zheng, Vahid Noroozi, and Philip S. Yu. 2017. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In *WSDM*. 425–434.

[49] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. In *ACL*.