Graph-based Extractive Explainer for Recommendations

A

Thesis

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

---

in partial fulfillment

of the requirements for the degree

Master of Science

by

Peng Wang

December 2021

APPROVAL SHEET

This

Thesis

is submitted in partial fulfillment of the requirements
for the degree of

Master of Science

Author: Peng Wang

This Thesis has been read and approved by the examing committee:

Advisor: Hongning Wang

Advisor:

Committee Member: Jundong Li

Committee Member: Yangfeng Ji

Committee Member:

Committee Member:

Committee Member:

Committee Member:

Accepted for the School of Engineering and Applied Science:

Jennifer L. West, School of Engineering and Applied Science

December 2021

To my parents, teachers and friends.

Acknowledgements

I would like to thank my advisor Professor Hongning Wang for his instruction and guidance on this project. We had a lot of great discussions about the model design and experiment results analysis via Slack and Zoom. Moreover, Professor Wang not only guided me through this project but also taught me how to do research. I am fortunate to have him as my advisor and I believe this research experience will be a precious and great starting point for my future study. I would also like to thank my thesis committee members, Professor Yangfeng Ji, and Professor Jundong Li for their feedback and suggestions in helping me further polish this work.

I would also like to thank Dr. Renqin Cai for his unreserved support of this project and tons of thoughtful discussions. As a senior Ph.D. student and also a friend of mine, he also gave me lots of suggestions and encouragement when I was perplexed and felt mixed-up about doing research. Without his support and help, I could not have reached my current accomplishments.

When I sat on the plane from China to the U.S. in 2019, I couldn't even imagine how drastically different my life would be in the next 2 years from what I expected. Taking most of my courses online makes my study at UVA a quite special experience for me. However, I never regret my decision of joining the Hoos' family as I have learnt a lot of advanced knowledge on computer science and also met many kind professors and fellow students here.

Last but not the least, I want to give my highest gratitude to my parents. I could not have gotten this far without their support.

Abstract

Explanations in a recommender system assist users make informed decisions among a set of recommended items. Great research attention has been devoted to generating natural language explanations to depict how the recommendations are generated and why the users should pay attention to the recommendation. However, due to the different limitations of those solutions, e.g., template-based or generation-based, it is hard to make the explanations easily perceivable, reliable and personalized at the same time.

In this work, we develop a graph attentive neural network model that seamlessly integrates user, item, attributes, and sentences for extraction-based explanation. The attributes of items are selected as the intermediary to facilitate message passing for user-item specific evaluation of sentence relevance. And to balance individual sentence relevance, overall attribute coverage, and content redundancy, we solve an integer linear programming problem to make the final selection of sentences. Extensive empirical evaluations against a set of state-of-the-art baseline methods on two benchmark review datasets demonstrated the generation quality of the proposed solution.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Nowadays, recommendations in online information service platforms, from e-commerce (such as Amazon and eBay) to streaming services (such as Netflix and youtube), have greatly shaped everyone's life, by affecting who sees what and when [1, 10, 22].

Therefore, besides improving the quality of recommendations, explaining the recommendations to the end users, e.g., how the recommendations are generated [35, 42, 29] and why the user should pay attention to the recommended content [39, 50, 46], is also critical to improve users' engagement and trust in the systems [2, 11, 31].

To be helpful in users' decision making, the system-provided explanations have to be easily **perceivable**, **reliable**, and **personalized**. Template-based explanations have been the dominating choice, where various solutions were developed to extract attribute keywords or attribute-opinion pairs from user reviews [46, 39, 50, 35] or from pre-existing knowledge graph [45, 42] to form the explanation content about a specific item for a target user. The fidelity of the generated explanations can be improved by careful quality control in the algorithms' input. But the predefined templates lack desirable diversity, and their rigid format and robotic style are less appealing to ordinary users [47, 16].

On the other hand, due to the encouraging expressiveness of the content generated from neural language models [9, 27, 4], an increasing number of solutions adopt generative models for explanation generation [47, 16, 17]. The generated content from such solutions are generally believed to have better readability and variability.

Table 1.1: Example explanations produced by 3 different types of explainable recommendation models.

Hotel	Kimpton Hotel Eventi, NYC
EFM	You might be interested in [staff/room], on which this hotel performs well.
SAER	The room was spacious and the room was clean. I was very pleased with the staff, the hotel and staff were very friendly.
GREENer	The view of the empire state building was incredible! The hotel was beautifully decorated, and the staff was very helpful. The room was huge and very clean, the bed comfy and the jacuzzi wonderful.
Hotel	New Orleans Marriott
EFM	You might be interested in [staff/location], on which this hotel performs well.
SAER	The location was great, the hotel was very nice, and the rooms were clean and comfortable. The room was spacious and the beds were extremely comfortable.
GREENer	There was a great view of the Mississippi river. Can't beat the location, just a few blocks from the heart of bourbon street.

Nevertheless, the high complexity of neural language models prevents fine-grained control in its generated content. And the success of such models heavily depends on the availability of large-scale training data. Due to the lack of observations about opinionated content from individual users on specific items, the generation from such models can hardly be personalized. On the contrary, it has been observed that such models' output tend to be generic and sometimes even less relevant to target items [47].

To understand the aforementioned advantages and limitations of these two types of explanation generation methods, we extract a few sample outputs from one typical solution of each type trained on the same hotel recommendation dataset in Table 1.1. We chose Explicit Factor Model (EFM) [50] to represent template-based solutions, and Sentiment Aligned Explainable Recommendation (SAER) [47] to represent neural generation based solutions. The robotic style of EFM's explanation content can be easily recognized, e.g., only the attribute keyword changes across its output for different items. Even if the target hotels in the example are indeed featured with staff or location, such generic content hurts the trustworthiness of the explanations. On the other hand, although SAER's output style is more diverse, its content is quite

generic; especially when across items, the aspects mentioned are less specific about the target items. This is also problematic when the user needs to choose from a set of recommended items based on their explanations.

To make the explanations easily perceivable, reliable, and also personalized, we propose an extractive solution, named **GR**aph **E**xtractive **E**xplai**Ner** (GREENer), to extract sentences from existing reviews for each user-item pair as explanations. By collectively selecting from review content, the extracted sentences maintain the readability from human-written content, and thus make the explanations easily perceivable. For a given pair of user and item, the past reviews from the user suggest his/her preferences on different aspects/attributes of this type of items; and the past reviews describing the item suggest its commonly received aspects. Hence, specificity about the user and item can be captured, which leads to personalized explanations. And because of the aggregation among user-provided content about the item, the reliability of the selected content can also be optimized.

Accurate extraction from existing content as explanations is however non-trivial. First, not all the sentences in a user review are relevant to the item. For example, it is very common to encounter users' personal experiences in a review. Such content is clearly unqualified as explanations and should be filtered. Second, the user and item should play different roles in selecting the sentences for explanation: the item suggests the set of relevant aspects, while the user suggests where the attention should be paid to. Therefore, the interplay between the user and item factors should be carefully calibrated when evaluating a sentence's relevance. Third, the selected sentences should cover distinct aspects of an item; and it is apparently undesirable to repeatedly mention the same aspect in different sentences when explaining an item. However, it is expected that an item's popular attributes will be mentioned in multiple users' reviews with some content variations. Avoiding such nearly duplicated content becomes necessary.

To address these challenges in extractive explanation generation, we develop a graph attentive neural network model that seamlessly integrates user, item, attributes and sentences for sentence selection. For a collection of items, we first extract fre-

quently mentioned attributes as the intermediary to connect users and items with sentences, i.e., the connectivity on the graph suggests who mentioned what about the item. As a result, sentences not related to any selected attributes are automatically filtered, which are clearly not qualified as explanations. To handle data sparsity when estimating the model parameters, we employ pre-trained language models [8] for attribute words' and sentences' initial encoding. Through attentive message passing on the graph, heterogeneous information from user, item and attributes about the sentences is aggregated for user-item specific evaluation of sentence relevance. However, because each sentence is independently evaluated by the neural network, content redundancy across sentences cannot be directly handled. We introduce a post-processing strategy based on Integer Linear Programming to select the final top-K output, where the trade-off between relevance and redundancy is optimized.

To investigate the effectiveness of GREENer for explanation generation, we performed extensive experiments on two large public review datasets, Ratebeer [47] and TripAdvisor [38]. Compared with state-of-the-art solutions for explanations, GREENer improved the explanation quality in both BLEU and ROUGE metrics. Our ablation analysis further demonstrated the importance of modeling these four types of information source for explanation generation, and also the importance of a graph structure for capturing the inter-dependency among them. Our case studies suggest that our produced explanations are more perceivable, specific to the target user-item pair, and thus more reliable.

Chapter 2

Related Work

Numerous studies have demonstrated that explanations play an important role in helping users evaluate results from a recommender system [32, 33, 49]. And various forms of explanations have been proposed, from social explanations such as "*X, Y and 2 other friends like this.*" [30], to item relation explanations such as "*A and B are usually purchased together.*" [20, 42, 45], and opinionated text explanations such as "*This* ***phone*** *is featured with its high-resolution* ***screen****.*" [50, 39, 47], which is the focus of this work.

There are currently three mainstream solutions to generate opinionated text content as explanations, namely template-based, generation-based, and extraction-based methods. They all work on user-provided item reviews to create textual explanations. In particular, template-based methods predict important attributes of the target item together with the sentiment keywords from user reviews to fill in the slots in those manually crafted templates. As typical solutions of this type, EFM [50] and MTER [39] predict important item attributes and corresponding user opinion words for a given recommendation via matrix factorization and tensor factorization. EX^3 extracts key attributes to explain a set of recommendations, based on the idea that the selected attributes should predict users' purchasing behavior of those items [46]. CountER employs counterfactual reasoning to select the important aspects for explanation [34]. The main focus in these template-based methods has been devoted to identify the most important item attributes and user opinion, i.e., to improve relia-

bility and personalization; but its lack of content variability and robotic explanation style make such explanations less appealing to the end users.

To increase content diversity in the provided explanations, neural language models are employed in generation-based methods to synthesize natural language explanations. As an earlier work, NRT models explanation generation and item recommendation with a shared user-item embedding space, where its predicted recommendation rating is used as part of the initial state for corresponding explanation generation [17]. NETE shared a very similar idea with NRT, but it further confines the generation to cover specific item attributes that are selected by a separated prediction module [16]. SAER constrains the sentiment conveyed in the generated explanation content to be close to the item's recommendation score [47]. However, due to the high complexity of neural language models, it is very hard to control such models' content generation at a fine granularity. As a result, the reliability of its generation is questionable. Furthermore, such methods tend to generate generic content to fit the overall data distribution in a dataset. Hence, on the user side, the explanation is less personalized; and on the item side, the explanation is even less relevant (e.g., overly generic).

Extraction-based solutions directly select representative sentences from the target item's existing reviews. And our proposed solution falls into this category. NARRE selects the most attentive reviews as the explanation, based on the attention originally learned to enrich the user and item representations for recommendation [7]. CARP uses the capsule network for the same purpose [15]. [44] adopt reinforcement learning to extract the most relevant review text that matches a given recommender system's rating prediction. In nature, extraction-based solutions model the affinity between user-item pairs with sentences, which is very sparse: a user review is typically short and a user often only writes one review for an item. Personalized explanation is hard to achieve in such a scenario. Our solution breaks this limitation by introducing item attributes as an intermediary, which not only alleviate sparsity issue but also improves specificity and reliability of the generated explanations (e.g., the sentences will only cover attributes associated with the target item). We should note that the

extraction-based solutions are restricted to an item's existing reviews; for items with limited exposure, e.g., a new item, these solutions (including ours) cannot provide any informative explanations. How to address this limitation is left as our future work.

Chapter 3

Methodology

In this section, we describe our proposed extractive explanation solution GREENer in detail. At the core of GREENer is a graph neural network, which integrates heterogeneous information about a user, a recommended item, and all candidate sentences via attentative message passing. To alleviate the observation sparsity issue at the user-item level, we introduce item attributes as an intermediary to connect user, item and sentences. The constructed graph structure is illustrated in Figure 3-1. Finally, the extraction is performed by solving an integer linear programming problem to balance individual sentence relevance, overall coverage and content redundancy in the selected sentences.

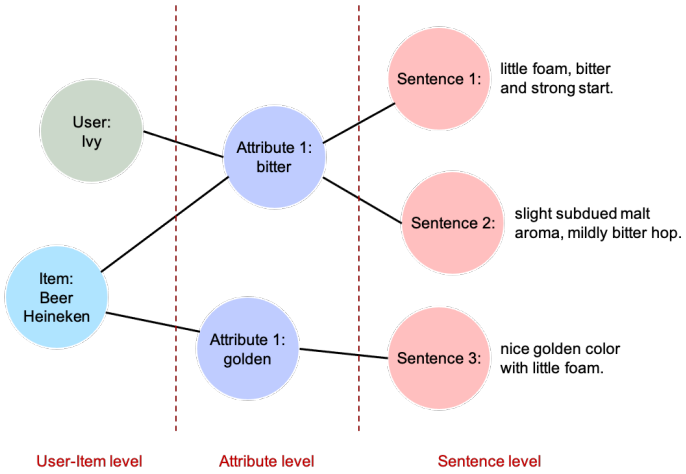

Figure 3-1: Illustration of the Graph model used by **GR**aph **E**xtractive **E**xplai**Ner** (GREENer). A heterogeneous graph is constructed for each user-item pair. The graph consists of 4 types of nodes from 3 different levels.

3.1 Problem Setup & Notations

We first define the notations employed in this paper. Denote the candidate recommendation item collection as $\mathcal{C}$ of size $|\mathcal{C}|$, the set of users as $\mathcal{U}$ of size $|\mathcal{U}|$, and the vocabulary of item attributes as $\mathcal{V}$ of size $|\mathcal{V}|$. We collect all the reviews from $\mathcal{U}$ about $\mathcal{C}$ and segment them into sentences. Then we denote $\mathcal{S}_{uc} = \{s^i_{uc}\}_{i=1}^N$ as all the sentences from user u about an item c, where each sentence $s_{uc} = \{w_i\}_{i=1}^T$ consists of T words $w \in \mathcal{V}$. We aggregate sentences from user u over all items into $\mathcal{S}_u = \{\mathcal{S}_{uc}\}_{c\in\mathcal{C}}$; and similarly, we aggregate sentences about item i from all users into $\mathcal{S}_c = \{\mathcal{S}_{uc}\}_{u\in\mathcal{U}}$. The attributes $\mathcal{F}$ are items' popular properties mentioned in the $\mathcal{D}$, which are a subset of $|F|$ words selected from the vocabulary $\mathcal{V}$. For a pair of a user $u \in \mathcal{U}$ and an item $c \in \mathcal{C}$, the goal of the model is to select K sentences $\{S^i\}_{i=1}^K$ from the union of sentences $\mathcal{S}_u \bigcup \mathcal{S}_c$, such that these K sentences best describe how the user would perceive the item.

Note that many widely used online product review datasets [23, 38] only provide rating and raw review text for each user-item pair. We extracted attributes $\mathcal{F}$ and sentences $\mathcal{S}_{uc}$ using NLP toolkits [3, 12, 50]. More details are introduced in section 4.1.

3.2 Neural Graph Model for Sentence Encoding

To produce explanations personalized to the pair of user and item, GREENer utilizes two modules to aggregate the pair of user and item information into the sentence representation. These two modules capture the dependence from different perspectives. To be more specific, one module explicitly constructs the structural relationship between user, item and sentences via a graph model and the other module implicitly connects user, item and sentences on the level of latent representations.

First, the past sentences written by the user, the past sentences describing the item, and the attributes appearing in sentences suggest the dependence between a sentence and the pair of user and item. To capture this contextualized (e.g., attributes, and other past sentences) dependence, GREENer utilizes the graph to model user,

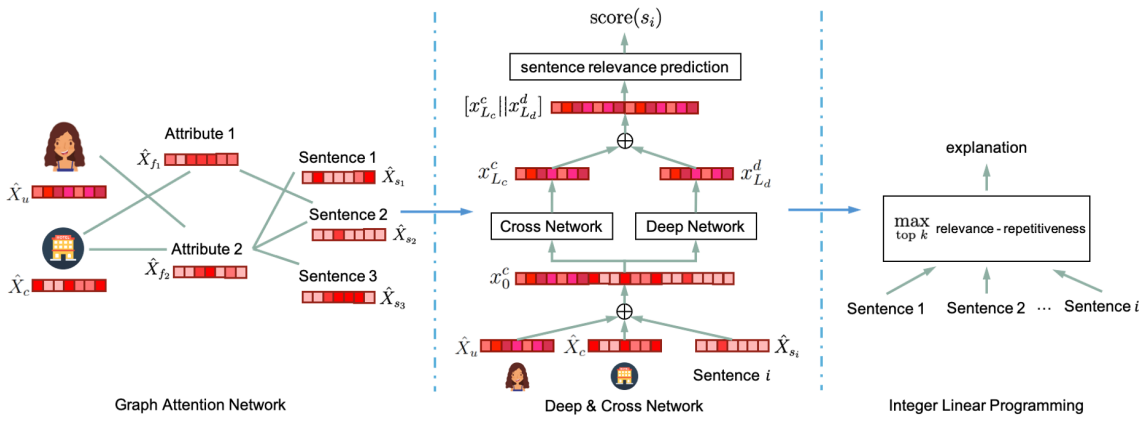

Figure 3-2: Overall illustration of GREENer. For a pair of user and item, GREENer utilizes graph attention network and deep cross network to encode past sentences written by the user and past sentences describing the item. Then it utilizes Integer Linear Programming to select sentences to synthesize an explanation.

item, attributes and sentences.

Second, the feature interactions among representations of user, item and sentences also suggest the dependence. To capture this self-dependence, GREENer utilizes the feature crossing to model user, item and sentences.

3.2.1 Graph Structure

Graph is a natural fit to model objects of multiple different types. Inspired by [37], GREENer utilizes graph to capture the contextualized dependence among user, item, attributes, and sentences. For each review written by a user towards an item, we construct a graph with the user, the item, sentences written by the user, sentences talking about the item, and attributes mentioned in these sentences, as illustrated in Figure 3-3 (and also Figure 3-2). To leverage the co-occurrence of user-attribute and item-attribute, in the graph, users are connected to attributes with edges and items are connected to attributes with edges. To propagate the personalized choices of attributes to the selection of sentences, the attributes are connected to sentences with edges. Feeding the graph into graph attention network, we obtain the sentence hidden representations and attribute hidden representations. Based on sentence hidden representations, we predict whether a sentence should be included as an explanation

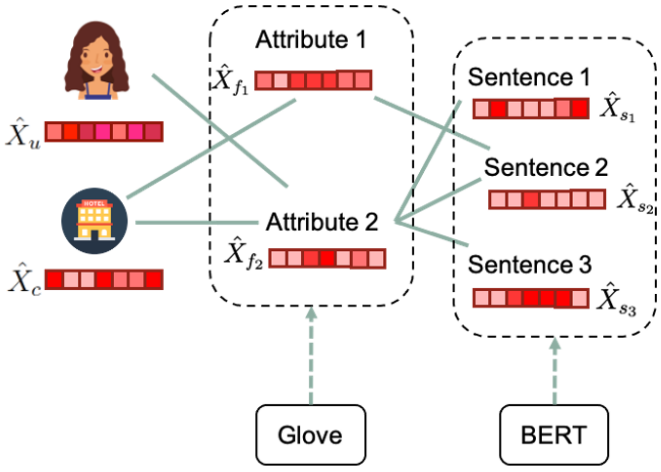

Figure 3-3: GAT module in GREENer.

or not. Likewise, based on attribute hidden representations, we predict whether an attribute should be mentioned in the explanation or not. The joint supervision from these two sources are used to train the model.

In the rest of this section, we will introduce our method to get the node's and edge's initial representations when we construct the graph, respectively. And in the next section, we will introduce how to learn the hidden node representations for user, item, attributes and sentences using a graph attention network (GAT).

Nodes on the Graph

A graph G consists of four types of nodes: a user node u, an item node c, $|S_g|$ sentence nodes where $S_g = S_u \bigcup S_c$ represents the union of sentences written by the user u and sentences talking about the item c, and M attribute nodes $\{f_g^i\}_{i=1}^M$ where $M \leq |F|$ appearing in sentences S_g. The set of sentences $\mathcal{S}_{uc} = \{s_{uc}^i\}_{i=1}^N$ written by the user u towards the item c is a subset of S_g, during training. While during testing, since $\mathcal{S}_u$ and $\mathcal{S}_c$ both come from the training set, $\mathcal{S}_{uc}$ isn't being included in S_g.

The input representation of the user node is a dense vector X_u, obtained by mapping the user index u through the input user embeddings $E_u \in R^{|U| \times d_u}$. Likewise, the input representation of the item node is a dense vector X_c, obtained by mapping the item index i through the input item embeddings $E_c \in R^{|C| \times d_c}$. In addition,

to represent attribute nodes, we map attributes $\{f_g^i\}_{i=1}^M$ into dense vectors $\{X_{f_g^i}\}_{i=1}^M$ through the input attribute embeddings $E_f \in R^{|F| \times d_f}$.

Because of great success in encoding and representing sentences achieved by the pre-trained large language models, i.e., Bidirectional Encoder Representations from Transformers (BERT) [8], we use BERT to encode sentences $\mathcal{S}_g$ as their input representations to get good semantic representations. Specifically, we first fine-tune the BERT on the corpus of reviews. Then we feed sentences into fine-tuned BERT to obtain their hidden vectors $\{X_{s_g^i}\}_{i=1}^{|S_g|}$ as input representations of sentence nodes.

Edges on the Graph

To capture the co-occurrence of user-attribute, we use an edge e_{uf} to connect user node u to attribute node f if the attribute has been used by the user (among all the reviews written by this user on the training set). Likewise, To capture the co-occurrence of item-attribute, we use an edge e_{cf} to connect item node c to attribute node f if the attribute has been used to describe the item (among all the reviews written for this item on the training set). To propagate the selection of attributes to the selection of sentences, we use an edge e_{fs} to connect attribute node f to sentence node s if the sentence contains the attribute word. Notice that all these edges are nondirectional. Through the path containing edges $e_{fs_i} \rightarrow e_{fs_j}$, the sentence s_i is connected to the sentence s_j. Consequently, the relationship between them are modeled. In this work, we only considered the binary edge weight (i.e. whether there is an edge between a pair of nodes or not) and did not consider other edge weights, which we leave as future work.

3.2.2 Graph Attention Layer

Given a constructed graph g for each user-item pair (u, c) with nodes X_u, X_c, X_f, X_s and edges e_{uf}, e_{cf}, e_{fs}, we adopt graph attention networks (GAT) [36] to encode co-occurrence information into node representations. Specifically, we stack L graph attention layer to map input node representations into output node representations

$\hat{X}_u, \hat{X}_c, \hat{X}_f, \hat{X}_s$. Due to the recursive nature of graph attention, we use a layer to illustrate the mechanism in our solution. For example, in the l-th layer, the inputs to the graph attention layer are $H^l = \{H_u, H_c, H_f, H_s\}$, which correspond to hidden representations of user node, item node, attribute nodes and sentence nodes, respectively. For the i-th node h_i^l in the graph, we obtain attention weights α^l of its connected nodes as,

$$\begin{aligned} z_{ij}^l &= \text{LeakyReLU}(W_a^l[W_q^l h_i^l || W_k^l h_j^l]) \\ \alpha_{ij}^l &= \frac{\exp(z_{ij}^l)}{\sum_{j' \in \mathcal{N}_i} \exp(z_{ij'}^l)} \end{aligned} \tag{3.1}$$

where $\mathcal{N}_i$ refers to other nodes connected to i-th node, i.e., the neighboring nodes of i-th node. W_a^l, W_q^l, W_k^l are trainable parameters and $||$ denotes the concatenation operation. As shown in Eq. 3.1, the attention weight between this pair of adjacent nodes is first computed based on the hidden representations and then being normalized by softmax.

With the attention weights, we obtain the output hidden representation of the i-th node in the l-th layer as

$$h_i^{l+1} = \sigma(\sum_{j \in \mathcal{N}_i} (\alpha_{ij}^l h_j^l)) \tag{3.2}$$

As suggested by the original GAT paper [36], using multi-head attention can stabilize the learning process and also strengthen the model capacity. With the d_h multi-head attention, we repeat the above process d_h times and merge the output hidden representations from d_h heads as the representation of the i-th node. For the intermediate layers, we merge d_h heads by concatenation:

$$h_i^{l+1} = ||_{head=1}^{d_h} h_{head,i}^{l+1} \tag{3.3}$$

where $h_{head,i}^{l+1}$ is obtained as Eq. 3.2. For the last layer, we merge d_h heads by average:

$$h_i^{l+1} = \frac{1}{d_h} \sum_{head=1}^{d_h} h_{head,i}^{l+1} \tag{3.4}$$

Note that for the initial attention layer, we use the input node representations X_u, X_c, X_f, X_s as the input H^0, and through L attention layers, we use the output representations H^L as the output node representations $\hat{X}_u, \hat{X}_c, \hat{X}_f, \hat{X}_s$. So far, we have already received the sentence node representation $\hat{X}_{s_i}$ (for the sentence s_i) which is specific for the user-item pair (u, c). We can then feed it into a linear layer as

$$\text{score}(s_i) = \langle W_o^s , \hat{X}_{s_i} \rangle \tag{3.5}$$

which predicts the relevance score that sentence s_i could be taken as an explanation sentence for the user-item pair (u, c).

3.2.3 Conditional Sentence Prediction and Feature Crossing

One remaining problem is that the final prediction layer in Eq. 3.5 is identical for all sentences, even if they are applied to different user-item pairs. Although $\text{score}(s_i)$ is still conditioned on user-item pair (u, c) since the same sentence s_i should have different output node representation $\hat{X}_{s_i}$ on different user-item pair (u, c), this influence is implicit. To emphasize personalization, we applied conditional sentence prediction, which starts from modifying Eq. 3.5 into

$$\text{score}(s_i) = \langle W_o^s , [\hat{X}_{s_i} || \hat{X}_u || \hat{X}_c] \rangle \tag{3.6}$$

Besides directly realizing the conditional sentence prediction by concatenating the output representation of sentence with corresponding user's and item's, GREENer seeks for more elaborate methods to integrate these information. Inspired by recent works in the field of Click Through Rate (CTR) prediction [6, 40, 41, 48], we leverage feature crossing, which is an effective way to model the feature interactions, to model

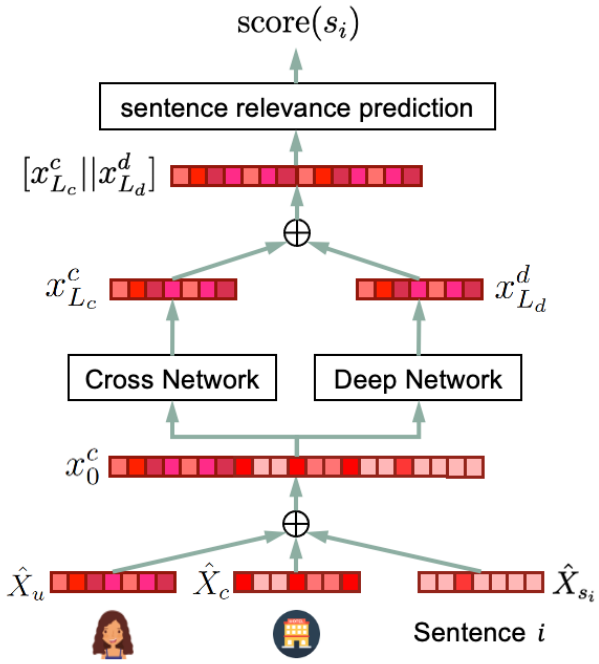

Figure 3-4: DCN module in GREENer.

the feature-level interaction among $\hat{X}_s$, $\hat{X}_u$ and $\hat{X}_c$.

GREENer utilizes the Deep & Cross network (DCN) [40] to apply feature crossing. DCN is a combination of Cross Network and Deep Network in a parallel structure. The Cross Network is a stack of multiple Cross Layers, which can be written as

$$x^c_{l+1} = x^c_0 x^{c\mathrm{T}}_l w^c_l + b^c_l + x^c_l \tag{3.7}$$

where x^c_l represents the hidden representation in l-th layer of the cross network. The hidden representation of $(l+1)$-th layer consists of the outer product of the input x^c_0 and previous layer's hidden representation x^c_l with a weight vector w^c_l, a bias term b^c_l and a residual term x^c_l from the l-th layer. Figure 3-5 shows one Cross Layer in the Cross Network.

The Deep Network is a multiple layer fully-connected neural network, which can be written as

$$x^d_{l+1} = f(W^d_l x^d_l + b^d_l) \tag{3.8}$$

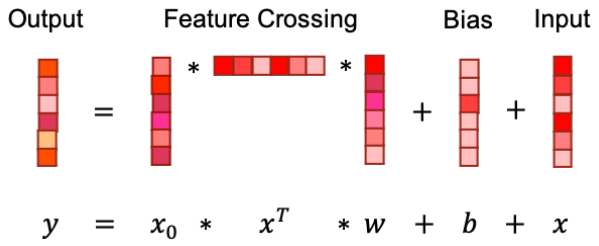

Figure 3-5: Illustration for one Cross Layer in the Cross Network of DCN module in GREENer.

where x_l^d represents the hidden state in l-th layer of the deep network. The hidden representation of $(l+1)$-th layer consists of the hidden representation x_l^d from the previous layer with a weight matrix W_l^d and a bias term b_l^d; and $f(\cdot)$ represents the ReLU function. $w_l^c, b_l^c, W_l^d, b_l^d$ are trainable parameters. These 2 networks takes the same input $x_0^c = x_0^d$ which is a concatenation of user, item and sentence output node representation from the GAT, as

$$x_0^c(s_i) = x_0^d(s_i) = [\hat{X}_u || \hat{X}_c || \hat{X}_{s_i}] \tag{3.9}$$

With this, we can rewrite Eq. 3.6 into:

$$\text{score}(s_i) = \langle W_o^s, [x_{L_c}^c(s_i) || x_{L_d}^d(s_i)] \rangle \tag{3.10}$$

where L_c and L_d are the number of layers of cross network and deep network, respectively; $x_{L_c}^c(s_i)$ and $x_{L_d}^d(s_i)$ are the outputs from the cross network and deep network of the i-th sentence, respectively.

3.3 Loss Function and Sentence Extraction

With the node representations outputted by the graph attention layer and subsequent Deep & Cross network, we make predictions of sentences being explanations. Considering that sentences in the candidate sentence set $\mathcal{S}_g$ but not in the ground-truth sentence set $\mathcal{S}_{uc}$ might also be very similar to the ground-truth sentences in

$\mathcal{S}_{uc}$, we assign each sentence in $\mathcal{S}_g$ with a soft label which measures its similarity with sentences in $\mathcal{S}_{uc}$ and utilize pairwise loss for sentence predictions. Considering that the task of predicting attributes appearing in explanations is related to the task of predicting sentences as explanations, we optimize these two tasks jointly by performing multi-task learning. Considering that the top-predicted sentences might consist of very similar content, a post-processing module which utilizes Integer Linear Programming (ILP) was added to GREENer so that the sentences being selected to synthesize the final explanation are both highly relevant and also less repetitive.

3.3.1 Pairwise Loss Function

Table 3.1 shows 2 pairs of examples from the TripAdvisor dataset [38], where the left-side column shows sentences selected in the candidate set $\mathcal{S}_g$ and the right-side column shows the sentence in the corresponding ground-truth sentence set $\mathcal{S}_{uc}$ which has very similar content with the sentence in the left-side column. Based on this observation that sentences that are not included in the ground-truth explanations can be also similar to sentences that are included in the ground-truth explanations, we first assign soft label for each sentence and then use pairwise loss as the loss function of sentence predictions.

Table 3.1: Example of semantic similar pairs of sentences from the candidate set $\mathcal{S}_g$ and the ground-truth sentences $\mathcal{S}_{uc}$.

candidate sentence	ground-truth sentence
the staff was very courteous and friendly.	the staff was very accommodating, pleasant and friendly.
the breakfast area was clean and stocked well.	the breakfast area was well stocked with food and we had plenty to eat.

Specifically, for each sentence, we obtain its similarities to every ground-truth sentence and use the maximum similarity as its relevance to the ground-truth. There are many possible choices for computing this similarity score, such as cosine similarity of the pair of sentences' BERT embeddings and the BLEU score [24] of the pair of sentences. Since GREENer is on sentence-level, we take the maximum similarity over each ground-truth sentence instead of the similarity between the candidate sentence

and the whole ground-truth review. Based on sentences' relevance, we obtain their pairwise order. Then we obtain the unnormalized scores of sentences being selected. With the pairwise order and unnormalized scores, we obtain the pairwise loss.

As there are some sentences in the candidate set $\mathcal{S}_g$ with very similar content, such as "*the staff was very* `courteous` *and friendly*" and "*the staff was very* `polite` *and friendly*", it's not worthy to differentiate them in the pairwise loss. Therefore, to promote the efficiency and better stabilize the learning process, we separate the relevance score into several buckets and assign discrete relevance to these buckets. For the i-th sentence, the discrete relevance score is denoted as r_i.

For the i-th sentence, recall that we can obtain its unnormalized score of being selected by Eq. 3.10, which take the concatenation of the outputs from the deep & cross network and then feed it into a linear layer as,

$$\text{score}(s_i) = \langle W_o^s, [x_{L_c}^c(s_i)||x_{L_d}^d(s_i)] \rangle$$

For the i-th sentence and the j-th sentence, the pairwise loss is then defined as,

$$L_s = -\sum_{i \in S_g} \sum_{j \in S_g} \text{sign}(r_i - r_j) \log \text{ sigmoid}(\text{score}(s_i) - \text{score}(s_j)) \tag{3.11}$$

$$\text{where sign}(r_i - r_j) := \begin{cases} 1 & \text{if } r_i > r_j, \\ 0 & \text{if } r_i == r_j, \\ -1 & \text{if } r_i < r_j. \end{cases}$$

3.3.2 Multi-task Loss Function

On one hand, although we do not directly differentiate a pair of sentences in the sentence-level pairwise loss if they possess very similar content, they can be distinguished at a more fine-grained level. For example, in the example we provided in the previous section, if the user turns out to mention the attribute "`courteous`" more often than the attribute "`polite`", the sentence "*the staff was very* `courteous` *and*

friendly" should be ranked higher than the other sentence. On the other hand, sentences with high relevance scores always have overlapping attributes with the ground-truth sentences. Therefore, besides sentence prediction loss, we also include the attribute prediction loss, to help GREENer learn better attribute nodes representations which will then help the model's learning procedure of the sentence nodes representations by message passing on the graph.

We obtain probabilities scores of attributes appearing in the explanation by feeding the output representations of attribute nodes $\hat{X}_f$ into a linear layer and then a sigmoid layer. For the i-th attribute node, its probability is obtained as,

$$p(f_i) = \text{sigmoid}(\langle W_o^f, \hat{X}_{f_i} \rangle) \tag{3.12}$$

We adopt the binary cross entropy as the loss function of attribute predictions. With the ground-truth label y_{f_i}, the loss of attribute predictions is,

$$L_f = -\sum_{i=1}^{M} y_{f_i} \log p(f_i) \tag{3.13}$$

Combining the sentence prediction loss from Eq. 3.11 and attribute prediction loss from Eq. 3.13, we obtain GREENer's objective function as

$$L = \lambda L_s + (1 - \lambda) L_f \tag{3.14}$$

where λ is a hyper-parameter to control the weight of each loss to the objective.

3.3.3 Synthesizing Explanation using ILP

As we assign each candidate sentence with relevance score that computed from some similarity metrics, assume we have a perfect model which output top-predicted sentences that are indeed very similar to the ground-truth sentences. The top-predicted sentences may still be similar to each other (e.g. talking about same attributes), which are not ideal for an informative explanation. Therefore, to reduce the repet-

itiveness in synthesized explanations, we select sentences that are dissimilar to each other. Thus, the optimization objective of selecting K sentences to synthesize an explanation is

$$\max \mathcal{O} = relevance(\{S_i\}_{i=1}^{K}) - \alpha \cdot repetitiveness(\{S_i\}_{i=1}^{K}) \tag{3.15}$$

where α handles the balance between relevance and repetitiveness. To find the optimal K sentences, we adopt Integer Linear Programming (ILP) to realize this optimization problem. Specifically, the objective function is expressed as,

$$\max \sum_{i=1}^{N} x_{s_i} \hat{r}_{s_i} - \alpha \sum_{i=1,j=i+1}^{i=N,j=N} y_{ij} \text{sim}(s_i, s_j) \tag{3.16}$$

$$s.t. \sum_{i=1}^{N} x_{s_i} = K \tag{3.17}$$

$$\sum_{i=1,j=i+1}^{i=N,j=N} y_{ij} = \frac{K * (K-1)}{2} \tag{3.18}$$

$$x_{s_i} + x_{s_j} \leq y_{ij} + 1, \ \forall \ \{i,j\}, j \geq i+1 \tag{3.19}$$

$$x_{s_i} \in \{0,1\}, \ \forall \ i, \tag{3.20}$$

$$y_{ij} \in \{0,1\}, \ \forall \ \{i,j\} \tag{3.21}$$

where $\hat{r}_{s_i}$ is the normalized predicted relevance score of the sentence i-th sentence s_i and $\text{sim}(s_i, s_j)$ represents the similarity score between sentence s_i and s_j. Eq. 3.16 is the objective function of the ILP which consists of 2 terms, i.e. the sum of the relevance scores of the selected sentences and the sum of the pairwise similarity scores of the selected sentences. These 2 terms are connected with a hyper-parameter α which is used to control the trade-off between relevance and repetitiveness. Eq. 3.17 restricts the ILP to select exactly K sentences. Eq. 3.18 and Eq. 3.19 together define the value of y_{ij} which is used to select legal pairwise similarity scores (i.e. only when s_i and s_j are both selected should their pairwise similarity score being included in the objective function). To be more specific, when $x_{s_i} = 1, \ x_{s_j} = 1$, we have $y_{ij} = 1$, when $x_{s_i} = 0, \ x_{s_j} = 1$, we have $y_{ij} = 0$, when $x_{s_i} = 1, \ x_{s_j} = 0$, we have $y_{ij} = 0$, and

when $x_{s_i}=0,\ x_{s_j}=0$, we have $y_{ij}=0$.

For the pairwise sentence similarity score, we can utilize the tf-idf representations of sentences or the dense vector representations of sentences derived from BERT to calculate the cosine similarities $\text{sim}(s_i, s_j)$ between two sentences.

There are also other methods to handle this redundancy reduction problem of the synthesized explanations such as Maximal Marginal Relevance (MMR) [5] and Trigram blocking [19, 43]. MMR greedily selects the next sentence with high relevance (i.e. high predicted score) and less redundancy (i.e. small cosine similarity score) with the already selected sentences. Trigram blocking is a simple version of MMR which greedily selects the next sentence based on its relevance and skip those sentences that have trigram overlapping with the already selected sentences. A formally expression of MMR (if applied in our problem setting) can be written as,

$$\text{MMR}(s_i) = \arg \max_{s_i \in \mathcal{S}_g \setminus \hat{\mathcal{S}}} [\beta \ \text{score}(s_i) - (1-\beta) \max_{s_j \in \hat{\mathcal{S}}} \text{sim}(s_i, s_j))]$$

where $\hat{\mathcal{S}}$ is the already selected sentences and β is a hyper-parameter that controls the balance between relevance and repetitiveness.

Our proposed ILP is an extension to the MMR and Trigram blocking methods. MMR and Trigram blocking are both greedy methods which means that the sentence with highest predicted score would be surely to be included in the synthesized explanation. While our ILP doesn't have this restriction and can then reach to a solution that might result in a higher objective value than the solution from MMR and Trigram blocking. Also, our ILP can be easily extended by adding other objectives and constraints. For example, besides the sentence-level relevance score, we can also include the attribute-level relevance score for each sentence by taking the sum of the attributes prediction scores for all the attributes in the selected sentences so that we are selecting sentences that are not only highly relevant but also containing correct attributes. In this work, we only focus on sentence-level relevance and repetitiveness and leave extending the ILP as future work.

3.4 Training & Testing

In this chapter, we have encountered $\mathcal{S}_g$ many times. It's worthy to clarify that $\mathcal{S}_g$ has tiny difference during training and testing. The sentences set $S_g = S_u \bigcup S_c$ in a graph as candidate sentences of an explanation refer to sentences appearing in the **training** set. During training, the candidate sentences include ground-truth sentences in explanations, while during testing the candidate sentences do not include ground-truth sentences (since $\mathcal{S}_{uc}$ is not in the training set).

During training, we follow Eq. 3.10 to obtain unnormalized scores and then use scores to obtain the pairwise loss. During testing, we follow Eq. 3.10 to obtain unnormalized scores and then we feed scores into sigmoid function to obtain probabilities of sentences being included in explanations. Finally, we apply ILP for each user-item pair to extract K sentences which are both highly related to this user-item pair and less repetitive among each other.

Chapter 4

Experiments

In this section, we investigate the effectiveness of our proposed solution GREENer on producing explanations. We conduct experiments on two large datasets. We compare our model against a set of state-of-the-art baselines to illustrate its advantage. In addition, we also did ablation analysis to study the importance of each components in GREENer.

4.1 Experiment Setup

Since GREENer focuses on producing explanations, in the experiments, we assume the recommended items are given. As GREENer is compatible with any recommendation algorithms, the recommended items can be provided by any recommendation algorithms. The datasets used for evaluation are Ratebeer dataset [21] and TripAdvisor dataset [38]. Both of them contain reviews crawled from corresponding platforms, including user, item, text content and rating information. In the Ratebeer dataset, the rating range is $[0, 20]$. Since recommender systems generally recommend items that are attractive to users, the desired explanations are positive sentiments. Thus, we use reviews with ratings larger than 10 to construct the corpus for experiments. In the TripAdvisor dataset, the rating range is $[1, 5]$. We use reviews with ratings larger than 3 to construct the corpus for experiments.

Table 4.1: Summary of the processed datasets. Rb stands for Ratebeer and TA stands for TripAdvisor.

	# Users	# Items	# Reviews	# Sentences	# Attributes
Rb	1,664	1,490	130,739	519,353	575
TA	4,954	4,493	287,879	1,001,235	462

4.1.1 Data Processing

Reviews have been directly used as explanations in many previous work [7, 44]. But as suggested in [23], a large portion of sentences in a review describes personal subjective experience, like "*I drank two bottles of this beer.*" and "*this was my first stay at this location.*", which could not provide any clues about the reason why the user like this item and hence they are not qualified as explanations. In contrast, sentences that serve as explanations should describe the properties of items to help users make informed decisions, like "*taste is of bubble gum and some banana.*" and "*close to a bus stop to ride into town.*". Therefore, we construct the explanation dataset by keeping informative sentences in the experiments. For both datasets, we use the Sentires toolkit [50] to extract attribute words from reviews and manually filter out inappropriate ones based on domain knowledge. Then, for each review, we only keep sentences that describe certain attributes of items as explanations.

We filter inactive users and unpopular items, i.e., we keep users who at least have written fifteen reviews and keep items which have been commented by users at least fifteen times. We keep the 20,000 most frequent words as the vocabulary and use "⟨unk⟩" token to represent others. The statistics of the processed datasets are reported in Table 4.1. We split the dataset into training, validation, and testing dataset according to the ratio 70%:15%:15%.

4.1.2 Baselines

We compare our model with five baselines, containing both generation-based methods and extraction-based methods, that can produce natural language sentences as explanations :

- **NRT**: Neural Rating and Tips Generation [17], a generation-based solution. It is originally proposed for tip (a short sentence summary) generation, but can be seamlessly adapted to generating explanations. It utilize the RNN language model to generate explanations.

- **SAER**: Sentiment Aligned Explainable Recommendation [47]. This is another generation-based solution. But it focuses specifically on the sentiment alignment between the rating and generated explanation. It implements a sentiment regularizer and a constrained decoding method to enforce the sentiment in the explanation to defend the rating in both training and inference phases.

- **NARRE**: Neural Attentional Regression model with Review-level Explanations [7]. It is an extraction-based solution. It learns the usefulness of the existing reviews through attention. Based on the attention over reviews, this model selects the review with largest attention value as the explanation.

- **SEER**: Synthesizing Aspect-Driven Recommendation Explanations from Reviews [14]. This is another extraction-based solution. It takes user's sentiment towards item's aspects as input, which can be obtained from explainable recommendation models such as EFM, and then form an ILP to select K most representative and coherent sentences that fits the user's demand of K aspects. Since SEER's result is controlled by the user-specific aspects which are not available in our problem setting, we implemented a variant which takes the top-k most popular attributes in the candidate sentence set $\mathcal{S}_g$ and use them as the user's demand aspects. We named this variant as SEER-$k$pop, where k can be any positive integer from 1 to $|\mathcal{F}|$.

- **ESCOFILT**: Unsupervised Extractive Summarization-Based Representations for Accurate and Explainable Collaborative Filtering[26]. This is also an extraction-based solution. It leverages on BERT's superior representation ability to generate user and item profile by clustering user-side and item-side sentences's corresponding embedding using K-Means. The K sentences that are nearest to

their own cluster centroids are selected to form the explanation.

4.1.3 Implementation Details

For both datasets, we first pre-trained 256-dimension Glove embeddings [25] on the whole vocabulary and fine-tune BERT model on our dataset using Sentence-BERT [28] which are later used to initialize the attribute node and sentence node, respectively. User and item node embedding size d_u and d_c are both set to 256. We stack 2 GAT layers, with 4 head in the first layer and 1 head in the second layer. The hidden size in GAT is 256. For DCN, we combine a 2 layer cross network with a 2 layer MLP whose hidden size and output size are both set to be 128. The sentence relevance score r_i used in pairwise loss is pre-computed maximum sentence BLEU score between s_i and every ground-truth sentences.

During training, we use a batch size of 16 and apply Adam optimizer [13] with a learning rate of 2e-4. The λ in multi-task loss function is set to be 0.5. The model is saved according to its performance on the valid set where we take the top-5 predicted sentences, connect them and compute the BLEU score with respect to the ground-truth review.

For synthesizing explanation during testing, we use the cosine similarity of tf-idf representations for a pair of sentences s_i and s_j to compute the pairwise similarity score $\text{sim}(s_i, s_j)$ in the ILP's objective function. Since the candidate sentences set $\mathcal{S}_g$ consists of both user-side sentence $\mathcal{S}_u$ and item-side sentences $\mathcal{S}_c$, sentence from the user-side might contain information that are counterfactual with the item's profile (for example, user-side sentences might taking about "*very nice and clean pool*" but this hotel doesn't even have a pool). To promote the reliability of the selected sentence, we filter out sentences that contain non-item-side attributes, i.e. if a sentence s_i contains attribute f_{ik} and f_{ik} doesn't appear in $\mathcal{S}_c$, then s_i will be removed from the candidate set. In order to reduce the computation complexity of the ILP, we select the top-100 predicted sentences for each user-item pair on the test set and use Gurobi[1] solver to select the top-5 most relevant yet not repetitive sentences to form the explanation.

[1]https://www.gurobi.com/products/gurobi-optimizer/

4.2 Quality of Generated Explanations

To comprehensively evaluate the quality of generated explanations, we measure the explanation quality with different types of metrics on different level of granularity. On the document-level, we utilize BLEU [24] score and ROUGE [18] score which are widely used in the area of machine translation and document summarization, respectively. On the attribute-level, we utilize the precision, recall and F1 score of the attributes contained in the selected sentences with respect to the ground-truth sentences. Document-level metrics are used to measure the perceivability of the synthesized explanation whereas attribute-level metrics are used to measure the personalization of the synthesized explanation.

4.2.1 Document-level Evaluations

Specifically, we utilize BLEU-{1, 2, 4} and F1 score of ROUGE-{1,2,L} to measure the precise and coverage of the selected sentences. We also included an additional metric (i.e. USR) to measure the personalization of the selected sentences at document-level. The results are reported in Table 4.2. Since GREENer generates 5 sentences for each user-item pair on both datasets, for the sake of fairness, we use the variant SEER-5pop which generates 5 sentences based on the 5 most popular attributes of each user-item pair. We also find that the generation-based methods such as NRT and SAER would generate quite similar and repetitive content among different user-item pairs if we use the greedy decoder. Thus, for NRT and SAER, we applied Top-5 sampling during decoding.

Analysis of the Performance on BLEU

GREENer outperforms baselines under every BLEU metric on both datasets. As BLEU is a precision-based metric, a larger BLEU achieved by a model suggests that a larger portion of content in sentences generated/selected by the model are relevant to the ground-truth sentences. Thus, the superior performance achieved by GREENer suggests that more content in sentences selected by GREENer can reflect the users'

Table 4.2: Comparison of document-level explanation quality by different models on Ratebeer and TripAdvisor.

Model	BLEU(%)			ROUGE(%)			UER
	1	2	4	1	2	L	
Ratebeer							
NRT	27.03	11.97	2.50	27.16	4.83	24.63	0.9971
SAER	28.40	12.68	2.66	27.59	4.92	25.29	0.9991
NARRE	24.67	9.09	1.41	22.13	2.79	20.04	0.1207
SEER-5pop	14.24	5.77	1.03	20.18	2.74	21.08	0.7882
ESCOFILT	26.36	12.27	3.55	27.55	5.58	24.62	**1.0**
GREENer	**36.20**	**18.49**	**5.80**	**32.80**	**7.92**	**29.32**	**1.0**
TripAdvisor							
NRT	20.30	8.31	1.70	20.25	2.92	18.86	0.9808
SAER	20.94	8.80	1.90	20.67	3.23	19.23	0.9734
NARRE	22.22	8.59	1.67	21.92	2.85	19.52	0.2488
SEER-5pop	21.84	9.00	1.89	21.77	3.13	20.44	0.7124
ESCOFILT	22.81	9.40	2.39	22.94	3.41	20.42	**1.0**
GREENer	**28.50**	**13.12**	**4.13**	**24.80**	**4.77**	**22.04**	0.9993

opinions towards the items. This shows the effectiveness of our model design.

The comparison of GREENer against NRT and SAER shows that the graph structure used in GREENer performs better in fusing users' preferences and items' properties into the produced sentences than the sequential structure modeled by RNN does. RNN-based methods such as NRT and SAER suffer from generating short and generic content. Generic content such as "*the staff was very friendly and helpful .*" indeed has larger chance to match with the ground truth. However, with the brevity penalty in BLEU, only generating 1 or 2 generic sentences eventually hurts the BLEU score.

The comparison of GREENer against NARRE shows that although both methods extract sentences from the corpus, graph structure used in GREENer can recognize sentences matching users' understandings of items more accurately than the attention structure used in NARRE can. Besides, compared to NARRE, GREENer extract sentences from a finer-grained level of sentence rather than the whole review, which results in higher flexibility on choosing sentences.

Both GREENer and SEER-5pop select the final K sentences using ILP. The com-

parison between these two methods shows that gradually learning sentence's relevance score performs better than directly modeling it from pre-computed user's sentiment scores over item's aspects. Since SEER-5pop has an objective to select sentences that are most representative on the demand aspect among the candidate set, it turns out to select short sentences such as "*dry finish.*" and "*good location.*", which explains its dropping performance on BLEU due to the brevity penalty.

The comparison between GREENer and ESCOFILT shows that GREENer's graph structure can capture user's preference on item's aspects and generate personalized explanations rather than semantically summarization of the item's profile. Although ESCOFILT's generated summarization may arguably cover more aspects about the target item, it's not always suitable from the user's perspective since he/she may only concentrate on a subset aspects of the item (this can also be verified from the result shown in Table 4.3) which explain the dropping on BLEU score comparing it with GREENer.

It's noteworthy that GREENer achieves much larger BLEU-4 than others do. This suggests the portion of 4-grams in sentences selected by GREENer overlapping 4-grams in ground-truth sentences is much larger than that in sentences selected by other models.

Analysis of the Performance on ROUGE

Moreover, GREENer outperforms baselines under every ROUGE metric on both datasets. As ROUGE is a recall-based metric, a larger ROUGE achieved by a model suggests that more content in ground-truth sentences are included in the sentences generated/selected by the model. Thus, the superior performance achieved by GREENer suggests that GREENer can recognize more content relevant to users about recommended items. This further demonstrates the effectiveness of GREENer. Especially GREENer achieves larger ROUGE-L than other models do. This indicates that GREENer is more powerful in identifying content with long consecutive words that match users' perceptions towards items from the corpus of sentences. The graph structure contributes to GREENer's better performance.

Analysis of the Performance on Personalization Metrics

To further analyze whether the explanations generated by GREENer is personalized to each user-item pair, we adopt an additional metrics, i.e. the ratio of unique explanations (UER). Following the definition in [16], the ratio of unique explanations is defined as,

$$\text{UER} = |E| \ / \ N_E$$

where E represents the set of generated unique explanations among all the generated explanations and N_E is the total number of generated explanations (i.e. the total number of user-item pair on the testing set).

Result shows that GREENer can reach a unique explanations ratio of nearly 1.0 on both datasets which is larger than both other generative-based methods and other extractive-based methods (except ESCOFILT). This shows that GREENer can indeed generated personalized explanation for each user-item pair. It's reasonable that ESCOFILT could reach a superior on this metric as it's a text summarization method that select most representative sentences among S_{uc}, which will be different for each user-item pair. However, its inferior performance on BLEU and ROUGE further indicates that simply viewing this explanation generation task as a text summarization task is not enough and modeling the user's preference on item attributes is essential.

4.2.2 Attribute-level Evaluations

To further measure the personalization of the synthesized explanation, we adopt attribute prediction metrics including precision, recall and F1 score of the attributes contained in the synthesized explanation with respect to the ground-truth explanation. The results are reported in Table 4.3.

GREENer outperforms baselines under Recall and F1 score of the attribute prediction on both datasets. Compared to generation-based methods including NRT and SAER, GREENer obtains similar precision score on both datasets, but significantly higher recall and F1. This indicates that although generation-based methods output

explanations with precise attributes, they contain fewer number of attributes (mainly generic ones) compared to GREENer. For extraction-based methods, SEER-5pop possesses larger precision score but smaller recall score which is reasonable as it's demanded to generate sentences about the top-5 most popular aspects for each user-item pair. ESCOFILT, on the other hand, possesses large recall score but smaller precision score which confirms our argument that ESCOFILT's generated summarization doesn't always contains the attributes that the user concentrated on.

Table 4.3: Comparison of attribute-level explanation quality by different models on Ratebeer and TripAdvisor. The best performing values are boldfaced and the second best are marked with underline.

Model	Attribute Prediction		
	P	R	F1
Ratebeer			
NRT	0.2927	0.2372	<u>0.2476</u>
SAER	0.2882	0.2375	0.2457
NARRE	0.2019	0.2438	0.2008
SEER-5pop	**0.3145**	0.2257	0.2417
ESCOFILT	0.2132	<u>0.3299</u>	0.2439
GREENer	<u>0.3063</u>	**0.3986**	**0.3283**
TripAdvisor			
NRT	<u>0.2282</u>	0.1922	0.1782
SAER	**0.2295**	0.1923	0.1802
NARRE	0.1732	0.2495	0.1821
SEER-5pop	0.2266	0.2615	<u>0.2195</u>
ESCOFILT	0.1692	<u>0.2821</u>	0.1926
GREENer	0.2062	**0.3208**	**0.2282**

4.3 Model Hyper-parameter and Structure Analysis

4.3.1 Hyper-parameter Tuning for ILP

Besides in GAT, hyper-parameters in ILP part can also significantly affect the final performance. For example, the number of sentences being extracted (i.e. k) and the balance between relevance and repetitiveness in ILP's objective function (i.e. α). We tuned k and α on the valid set of Ratebeer and the results are shown in Table 4.4.

According to the result, we choose $k = 5$ and $\alpha = 2.0$.

Table 4.4: Tuning hyper-parameters used in ILP

Parameter	BLEU(%)			ROUGE(%)		
	1	2	4	1	2	L
Tuning k						
$k = 1$	2.32	1.29	0.47	18.37	5.49	18.42
$k = 3$	25.47	13.38	4.56	**30.26**	**7.95**	27.61
$k = 5$	**35.07**	**18.03**	**5.81**	32.07	**7.95**	**28.92**
$k = 10$	23.84	12.66	4.23	28.98	7.49	28.18
Tuning α						
$\alpha = 0.0$	28.58	15.11	5.74	25.40	7.50	26.26
$\alpha = 0.5$	34.73	17.91	5.79	32.05	**8.02**	**28.92**
$\alpha = 1.0$	35.07	18.03	5.81	**32.07**	7.95	**28.92**
$\alpha = 2.0$	**35.25**	**18.11**	**5.83**	**32.07**	7.93	**28.92**

4.3.2 Ablation Analysis

We include four variants of our solution to study the contribution of each component to the superior performance of GREENer:

- ¬ *GAT.* This variant excludes the graph model and only preserves the user, item, feature and sentence information. For each sentence $\mathcal{S}_{uc}$, we concatenate its embedding with user u, item c and mean-pooling of $f_{\mathcal{S}_{uc}}$, where $f_{\mathcal{S}_{uc}}$ represents the features in sentence $\mathcal{S}_{uc}$. Then the concatenated embedding is directly fed into the DCN to predict the relevance score of sentence $\mathcal{S}_{uc}$. The comparison between this variant and GREENer indicates the advantage of modeling the user, item, feature and sentence into graph and using GAT to learn their intermediate relationships.

- ¬ *BERT.* This variant replaces BERT with the average words embeddings for sentence representation. The comparison between this variant and GREENer demonstrates the importance of using BERT to encode sentences as input sentence node representations.

- ¬ *DCN*. This variant replaces DCN in GREENer with a single linear layer. The comparison between this variant and GREENer presents the effectiveness of performing conditional sentence prediction by modeling both low-order and high-order feature interaction among user, item and sentence.

- ¬ *ILP*. This variant replaces the ILP post-processing decoding strategy with the vanilla strategy which we select sentences solely based on predicted probabilities of sentences in descending order. The comparison between this variant and GREENer shows the utility of considering the both relevance and repetitiveness when selecting sentences as explanations.

Table 4.5: Ablation analysis on Ratebeer dataset.

Model	BLEU (%)			ROUGE (%)		
	1	2	4	1	2	L
Ratebeer						
GREENer	**36.20**	**18.49**	**5.80**	**32.80**	**7.92**	**29.32**
¬ GAT	32.93	17.21	5.57	31.25	7.78	28.71
¬ BERT	33.47	16.61	4.99	31.73	7.29	28.40
¬ DCN	35.21	17.71	5.30	31.98	7.37	28.89
¬ ILP	28.58	15.11	5.74	25.40	7.50	26.26

Results of ablation analysis between GREENer are its 4 variants are reported in Tabel 4.5. Same as Table 4.2, we report BLEU-{1, 2, 4} and F1 score of ROUGE-{1,2,L}. All these variants perform worse than GREENer. Comparison between GREENer and the variant without GAT suggests that modeling the graph structure indeed help the model to learn the relationship between user's preference and item's aspects which cannot be easily tackled by directly performing feature crossing. In addition, using BERT to obtain sentence representations enables GREENer to make use of the co-occurrence of feature-sentence to recognize relevant sentences. Moreover, although the variant without DCN can already reach a promising result, combining DCN with GAT can further boost the performance which indicates that explicitly emphasizing user and item by feature crossing during the sentence prediction stage helps the model to choose sentences that better reveal the user's preference and item's attributes. Finally, ILP plays a significant role on extracting sentences that are both

relevance and less repetitive which is crucial to build a perceivable and trustworthy explainable recommendation system.

4.4 Case Study

We present two group of example explanations produced by GREENer and other baselines in Table 4.6. The ground-truth explanations are also included for reference. We manually labeled the overlapping attributes in the generated sentences for the 3 methods. From the table, we could see that the extracted sentences by GREENer are close to the ground truth explanations. The features in extracted sentences match those in ground truth explanations, especially some uncommon attributes such as "roasty chocolate" in example 1 and "hot tub" in example 2. With these features, the explanation generated by GREENer can help the user have a clear personalized and informative understanding of the item.

Table 4.6: Example explanations produced by different models on Ratebeer and TripAdvisor.

Model	Explanation
Ratebeer Example	
Ground-Truth	roasty , chocolate malts paired with chinook hops .
	very smooth sipper with a nice balance between sweet , smooth malt , and tangy hops .
NARRE	medium carbonation and body ; with a very nice creamy and slightly slick mouthfeel .
SAER	a dark orange color with a medium - sized white head .
GREENer	tastes of **hops** and **roasty chocolate** .
	taste is very **smooth** with bitter and herbal **hops** with creamy toasted **malts** .
TripAdvisor Example	
Ground-Truth	the rooms are the perfect size and have everything you would need .
	the outdoor hot tub is also fantastic for your aching legs after a full day of skiing .
NARRE	the hotel is very centrally located so we were able to walk which was really nice .
SAER	great hotel for breakfast , and the staff are very friendly . **room** is clean and spacious .
GREENer	the **hot tub** is great and a good place to meet people .
	comfortable beds , nice large bathrooms . maybe the cleanest motel **room** i 've ever been in .

Chapter 5

Conclusion and Future Work

In this paper, we present a graph neural network based extractive solution for explaining a system's recommended items to users. It integrates heterogeneous information about user, item, attributes and candidate sentences to evaluate the relevance of a sentence with respect to a particular user-item pair. Item attributes are introduced as the intermediary to address sparsity in observations at the user-item level, and for the same purpose pre-trained language models are used to encode item attributes and sentences for the model. Finally, to optimize the trade-off among individual sentence relevance, overall attribute coverage and content redundancy, we solve an integer linear programming problem to make the final selection of sentences for a user-item pair. Extensive experiment comparisons against a set of state-of-the-art explanation methods demonstrate the advantages of our solution in providing high-quality explanation content.

Extraction-based explanation methods still have their intrinsic limitations: their input is restricted to an item's existing reviews; for items with limited exposure, e.g., a new item, such solutions (including ours) cannot provide any informative explanations. Our current attempt to address this limitation was to incorporate the same user's historical reviews on items from the same category. Leveraging generative solutions to synthesize explanations could be a potential choice in such situations. Besides, as we mentioned in section 3.2.1, our current graph model didn't take into account continuous edge weight (such as tf-idf value) which actually should also be

useful to represent the attentive relationships between user-attribute, item-attribute and attribute-sentence other than our current method which solely based on node representations. Moreover, although we removed reviews which have negative or neutral ratings, there are still some sentences that convey negative sentiments. If these sentences are provided to a user-item pair which indeed have a high rating, the reliability of the selected sentences might be a concern. As our current model didn't explicitly model the sentiment between user and item, an additional constraint could be added in the ILP so that only the sentences that align with the demanded sentiment should be selected. In addition, currently the scoring function weights in our ILP was manually tuned. We can consider learning-based methods to optimize it for better performance in our future work.

Bibliography

[1] Charu C Aggarwal et al. *Recommender systems*, volume 1. Springer, 2016.

[2] Mustafa Bilgic and Raymond J Mooney. Explaining recommendations: Satisfaction vs. promotion. In *Beyond Personalization Workshop, IUI*, volume 5, 2005.

[3] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[4] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[5] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336, 1998.

[6] Bo Chen, Yichao Wang, Zhirong Liu, Ruiming Tang, Wei Guo, Hongkun Zheng, Weiwei Yao, Muyu Zhang, and Xiuqiang He. Enhancing explicit and implicit feature interactions via information sharing for parallel deep ctr models. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3757–3766, 2021.

[7] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. Neural attentional rating regression with review-level explanations. In *Proceedings of the 2018 World Wide Web Conference*, pages 1583–1592, 2018.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[9] Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge. *Computer Speech & Language*, 59:123–156, 2020.

[10] Daniel Fleder and Kartik Hosanagar. Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. *Management science*, 55(5):697–712, 2009.

[11] Jonathan L Herlocker, Joseph A Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250. ACM, 2000.

[12] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

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

[14] Trung-Hoang Le and Hady W Lauw. Synthesizing aspect-driven recommendation explanations from reviews. IJCAI, 2021.

[15] Chenliang Li, Cong Quan, Li Peng, Yunwei Qi, Yuming Deng, and Libing Wu. A capsule network for recommendation and explaining what you like and dislike. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–284, 2019.

[16] Lei Li, Yongfeng Zhang, and Li Chen. Generate neural template explanations for recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 755–764, 2020.

[17] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. Neural rating regression with abstractive tips generation for recommendation. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 345–354, 2017.

[18] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[19] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China, November 2019. Association for Computational Linguistics.

[20] Weizhi Ma, Min Zhang, Yue Cao, Woojeong Jin, Chenyang Wang, Yiqun Liu, Shaoping Ma, and Xiang Ren. Jointly learning explainable rules for recommendation with knowledge graph. In *The World Wide Web Conference*, pages 1210–1221, 2019.

[21] Julian McAuley, Jure Leskovec, and Dan Jurafsky. Learning attitudes and attributes from multi-aspect reviews. In *2012 IEEE 12th International Conference on Data Mining*, pages 1020–1025. IEEE, 2012.

[22] Silvia Milano, Mariarosaria Taddeo, and Luciano Floridi. Recommender systems and their ethical challenges. *AI & SOCIETY*, 35(4):957–967, 2020.

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

[24] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[25] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[26] Reinald Adrian Pugoy and Hung-Yu Kao. Unsupervised extractive summarization-based representations for accurate and explainable collaborative filtering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2981–2990, 2021.

[27] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[28] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[29] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[30] Amit Sharma and Dan Cosley. Do social explanations work?: studying and modeling the effects of social explanations in recommender systems. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1133–1144, 2013.

[31] Rashmi Sinha and Kirsten Swearingen. The role of transparency in recommender systems. In *CHI'02 extended abstracts on Human factors in computing systems*, pages 830–831. ACM, 2002.

[32] Kirsten Swearingen and Rashmi Sinha. Beyond algorithms: An hci perspective on recommender systems. In *ACM SIGIR 2001 Workshop on Recommender Systems*, volume 13, pages 1–11. Citeseer, 2001.

[33] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. Providing justifications in recommender systems. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 38(6):1262–1272, 2008.

[34] Juntao Tan, Shuyuan Xu, Yingqiang Ge, Yunqi Li, Xu Chen, and Yongfeng Zhang. Counterfactual explainable recommendation. *arXiv preprint arXiv:2108.10539*, 2021.

[35] Yiyi Tao, Yiling Jia, Nan Wang, and Hongning Wang. The fact: Taming latent factor models for explainability with factorization trees. In *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 295–304, New York, NY, USA, 2019. ACM.

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

[37] Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. Heterogeneous graph neural networks for extractive document summarization. *arXiv preprint arXiv:2004.12393*, 2020.

[38] Hongning Wang, Yue Lu, and Chengxiang Zhai. Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 783–792, 2010.

[39] Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. Explainable recommendation via multi-task learning in opinionated text data. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 165–174, 2018.

[40] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*, pages 1–7. 2017.

[41] Ruoxi Wang, Rakesh Shivanna, Derek Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the Web Conference 2021*, pages 1785–1797, 2021.

[42] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5329–5336, 2019.

[43] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The World Wide Web Conference*, pages 2022–2032, 2019.

[44] Xiting Wang, Yiru Chen, Jie Yang, Le Wu, Zhengtao Wu, and Xing Xie. A reinforcement learning framework for explainable recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 587–596. IEEE, 2018.

[45] Yikun Xian, Zuohui Fu, Shan Muthukrishnan, Gerard De Melo, and Yongfeng Zhang. Reinforcement knowledge graph reasoning for explainable recommendation. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 285–294, 2019.

[46] Yikun Xian, Tong Zhao, Jin Li, Jim Chan, Andrey Kan, Jun Ma, Xin Luna Dong, Christos Faloutsos, George Karypis, Shan Muthukrishnan, et al. Ex3: Explainable attribute-aware item-set recommendations. In *Fifteenth ACM Conference on Recommender Systems*, pages 484–494, 2021.

[47] Aobo Yang, Nan Wang, Hongbo Deng, and Hongning Wang. Explanation as a defense of recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 1029–1037, 2021.

[48] Weinan Zhang, Jiarui Qin, Wei Guo, Ruiming Tang, and Xiuqiang He. Deep learning for click-through rate estimation. *arXiv preprint arXiv:2104.10584*, 2021.

[49] Yongfeng Zhang and Xu Chen. Explainable recommendation: A survey and new perspectives. *arXiv preprint arXiv:1804.11192*, 2018.

[50] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 83–92, 2014.