Comparative Explanations of Recommendations

Aobo Yang AY6GV@VIRGINIA.EDU
Department of Computer Science
University of Virginia
Charlottesville, VA 22903, USA

Nan Wang NW6A@VIRGINIA.EDU
Department of Computer Science
University of Virginia
Charlottesville, VA 22903, USA

Renqin Cai RC7NE@VIRGINIA.EDU
Department of Computer Science
University of Virginia
Charlottesville, VA 22903, USA

Hongbo Deng DHB167148@ALIBABA-INC.COM
Alibaba Group
Hangzhou, China

Hongning Wang HW5X@VIRGINIA.EDU
Department of Computer Science
University of Virginia
Charlottesville, VA 22903, USA

Abstract

As recommendation is essentially a *comparative (or ranking)* process, a good explanation should illustrate to users why an item is believed to be better than another, i.e., comparative explanations about the recommended items. Ideally, after reading the explanations, a user should reach the same ranking of items as the system's. Unfortunately, little research attention has yet been paid on such comparative explanations.

In this work, we develop an extract-and-refine architecture to explain the relative comparisons among a set of ranked items from a recommender system. For each recommended item, we first extract one sentence from its associated reviews that best suits the desired comparison against a set of reference items. Then this extracted sentence is further articulated with respect to the target user through a generative model to better explain why the item is recommended. We design a new explanation quality metric based on BLEU to guide the end-to-end training of the extraction and refinement components, which avoids generation of generic content. Extensive offline evaluations on two large recommendation benchmark datasets and serious user studies against an array of state-of-the-art explainable recommendation algorithms demonstrate the necessity of comparative explanations and the effectiveness of our solution.

Keywords: explainable recommendation, comparative explanation, text generation, extract-and-refine

1. Introduction

Modern recommender systems fundamentally shape our everyday life (Koren et al., 2009; He et al., 2017; Sarwar et al., 2001; Rendle, 2010; Aggarwal et al., 2016). As a result, how to explain the algorithm-made recommendations becomes crucial in building users' trust in the systems (Zhang and Chen, 2020). Previous research shows that explanations, which illustrate how the recommendations are generated (Ribeiro et al., 2016; Lundberg and Lee, 2017) or why the users should pay attention to the recommendations (Wang et al., 2018a; Sun et al., 2020; Yang et al., 2021), can notably strengthen user engagement with the system and better assist them in making informed decisions (Bilgic and Mooney, 2005; Herlocker et al., 2000; Sinha and Swearingen, 2002).

Figure 1: An illustration about the necessity of comparative explanations. The recommended Hotel A, B, C are listed in a descending order, with the provided explanations to justify the ranking. But if we replace Hotel C's explanation with the one in the dash box, users may no longer perceive the ranking of all three hotels.

When being presented with a list of recommendations, typically sorted in a descending order, a user needs to decide which recommendation is his/her best choice. In other words, the provided explanations should help users *compare* the recommended items. Figure 1 illustrates the necessity of comparative explanations. As shown in the figure, by reading the explanations for the three recommended hotels, one can easily tell why the system ranks them in such an order. But if the system provided the explanation in the dashed box for Hotel C, it would confuse the users about the ranking, e.g., Hotel C becomes arguably comparable to top ranked Hotel A; but it was ranked at the bottom of the list. This unfortunately hurts users' trust in all three recommended hotels.

Existing explainable recommendation solutions are not optimized to help users make such comparative decisions for two major reasons. First, the explanation of a recommended item is often independently generated without considering other items in the recommendation list. As shown in Figure 1, one single low-quality generation (the one in the dashed box) might hamper a user's understanding over the entire list of recommendations. Second, the popularly adopted neural text generation techniques are known to be flawed of its generic content output (Holtzman et al., 2019; Welleck et al., 2019). Particularly, techniques like maximum likelihood training and sequence greedy decoding lead to short and repetitive sentences composed of globally frequent words (Weston et al., 2018). Such generic content cannot fulfill the need to differentiate the recommended items. Consider the example shown in Figure 1 again, "the hotel is good" is a very generic explanation and thus not informative. Its vague description (e.g., the word "good") and lacks of specificity (e.g., the word "hotel") make it applicable to many hotels, such that users can hardly tell the relative comparison of the recommended items from such explanations.

In this work, we tackle the problem of comparative explanation generation to help users understand the comparisons between the recommended items. We focus on explaining how one item is compared with another; then by using a commonly shared set of items as references (e.g., items the user has reviewed before), the comparisons among the recommended items emerge. For example, if the explanations suggest item A is better than item B and item C is worse than item B, the comparison between A and C is apparent after reading the associated explanations. Since there are already plenty of effective recommendation algorithms deployed in practice, we decide not to invent yet another. Instead, we assume the existence of a performing recommendation algorithm, and build our solution on top of its provided item rankings. We do not have any assumptions about how this recommender system ranks items (e.g., collaborative filtering(Sarwar et al., 2001) or content-based (Balabanović and Shoham, 1997)), but only require it to provide a ranking score for each item to our model (i.e., ordinal ranking) and this ranking score reflects a user's preference over the recommended items. This makes our solution generic and readily applicable to explain most existing recommendation algorithms.

We design an extract-and-refine text generation architecture to explain the ranked items one at a time to the user, conditioned on their recommendation scores and associated review text content. We refer to the item to be explained in the ranked list as the target item, and user we are explaining to as the target user. First, the model extracts one sentence from the existing review sentences about the target item as a prototype, with a goal to maximize the likelihood of fitting the intended opinion of the target user with respect to his/her writing style of comparative content, which is reflected in all review sentences written by this user before. Then we refine the extracted prototype sentence through a generative model to further improve its explanation quality (e.g., informativeness, fluency, and diversity of content). In this two stage procedure, the extraction module directly exploits the content already provided about the target item to ensure the relevance of generated explanations (e.g., avoid mentioning features that do not exist in the target item); and the refinement module further polishes the wording and sentiment for the target user. We design a new explanation quality metric based on BLEU to guide the end-to-end training of the extraction and refinement module, with a particular focus to penalize short and generic content in generated explanations.

We compared the proposed solution with a rich set of state-of-the-art baselines for explanation generation on two large-scale recommendation datasets: RateBeer (McAuley et al., 2012) and TripAdvisor (Wang et al., 2010). Besides, we also conducted extensive user studies to have the generated explanations evaluated by real users. Positive results obtained on both offline datasets and online user studies suggested the effectiveness of comparative explanations in assisting users better understand the recommendations and make more informed choices.

2. Related Work

Most explainable recommendation solutions exploit user reviews as the source of training data. They either directly extract from reviews or synthesize content to mimic the reviews. Extraction-based solutions directly select representative text snippets from the target item's existing reviews. For example, NARRE (Chen et al., 2018) selects the most attentive reviews as the explanation, based on the attention that is originally learned to enrich the user and item representations for recommendation. CARP (Li et al., 2019) uses the capsule network for the same purpose. (Wang et al., 2018b) adopt reinforcement learning to extract the most relevant review text that matches a given recommender system's rating prediction. (Xian et al., 2021) extract attributes from reviews to explain a set of items based on users' preferences. However, as such solutions are restricted to an item's existing reviews, their effectiveness is subject to the availability and quality of existing content. For items with limited exposure, e.g., a new item, these solutions can hardly provide any informative explanations.

Generation-based solutions synthesize textual explanations that are not limited to existing reviews. One branch focuses on predicting important aspects of an item (such as item features) from its associated reviews as explanations (Wang et al., 2018a; Tao et al., 2019; He et al., 2015; Ai et al., 2018). For instance, MTER (Wang et al., 2018a) and FacT (Tao et al., 2019) predict item features that are most important for a user to justify the recommendation. They rely on predefined text templates to deliver the predicted features. The other branch applies neural text generation techniques to synthesize natural language sentences. In particular, NRT (Li et al., 2017) models item recommendation and explanation generation in a shared user and item embedding space. It uses its predicted recommendation ratings as part of the initial state for explanation generation. MRG (Truong and Lauw, 2019) integrates multiple modalities from user reviews, including ratings, text, and associated images, for multi-task explanation modeling.

Our work is closely related to two recent studies, DualPC (Sun et al., 2020) and SAER (Yang et al., 2021), which focus on strengthening the relation between recommendations and explanations. Specifically, DualPC introduces duality regularization based on the joint probability of explanations and recommendations to improve the correlation between recommendations and generated explanations. SAER introduces the idea of sentiment alignment in explanation generation. However, both of them operate in a *pointwise* fashion, i.e., independent explanation generation across items. Our solution focuses on explaining the comparisons between items. We should also emphasize our solution is to explain the comparison among a set of recommended items, rather than to find comparable items (McAuley et al., 2015; Chen et al., 2020).

There are also solutions exploiting other types of information for explainable recommendation, such as item-item relation (Chen et al., 2021), knowledge graph (Xian et al., 2019) and social network (Ji and Shen, 2016). But they are clearly beyond the scope of this work.

3. Comparative Explanation Generation

Item recommendation in essence is a ranking problem: estimate a recommendation score for each item under a given user and rank the items accordingly, such that the utility of the recommendations can be maximized (Rendle et al., 2012; Karatzoglou et al., 2013). Instead of explaining how the recommendation scores are obtained, our work emphasizes on explaining how the comparisons between the ranked items are derived.

To learn the explanation model, we assume an existing corpus of item reviews from the intended application domain (e.g., hotel reviews). Each review is uniquely associated with a user u and an item c, and a user-provided rating r_c^u suggesting his/her opinion towards the item. We group the reviews associated with user u to construct his/her profile $\Omega_u = \{(x_1^u, r_1^u), (x_2^u, r_2^u), ..., (x_m^u, r_m^u)\}$, where x_i^u is the i-th review sentence extracted from user u's reviews and r_i^u is the corresponding opinion rating. r_i^u can be easily obtained when the detailed aspect ratings are available (Wang et al., 2010); otherwise off-the-shelf sentiment analysis methods can be used for the purpose (interested users can refer to (Wang et al., 2018a; Zhang et al., 2014) for more details). We create the item profile as $\Psi_c = \{x_1^c, x_2^c, ..., x_n^c\}$, where x_j^c is the j-th review sentence extracted from item c's existing reviews. Unlike the user profile, the item profile does not include ratings. This is because the ratings from different users are not directly comparable, as individuals understand or use the numerical ratings differently. Our solution is agnostic to the number of entries in user profile Ω_u and item profile Ψ_c in each user and item.

We impose a generative process for a tuple (x, r_c^u) from user u about item c conditioned on Ψ_c and Ω_u. We assume when user u is reviewing item c, he/she will first select an existing sentence from Ψ_c that is mostly related to the aspect he/she wants to cover about the item. Intuitively, this can be understood as the user will first browse existing reviews of the item to understand how the other users evaluated this item. Then he/she will rewrite this selected sentence to reflect his/her intended opinion and own writing style. This can be considered as *a set to sequence generation problem*. For our purpose of explanation generation, we only concern the generation of opinionated text x. Hence, we take opinion rating r_c^u as input, which leads us to the following formulation,

$$P(x|u, c, r_c^u) = \sum_{x_j^c \in \Psi_c} P_{ref}(x|x_j^c, r_c^u, \Omega_u) P_{ext}(x_j^c|r_c^u, \Omega_u) \tag{1}$$

where $P_{ext}(x_j^c|r_c^u, \Omega_u)$ specifies the probability that x_j^c from item profile Ψ_c will be selected by user u, and $P_{ref}(x|x_j^c, r_c^u, \Omega_u)$ specifies the probability that user u will rewrite x_j^c into x. We name the resulting model Comparative Explainer, or CompExp in short.

In Eq Eq (1), $P_{ext}(x_j^c|r_c^u, \Omega_u)$ is essential to capture the comparative textual patterns embedded in user u's historical opinionated text content. To understand this, we can simply rewrite its condition part: define $\Delta r_i^u = r_c^u - r_i^u$, we have $(r_c^u, \Omega_u) = \{(x_i^u, \Delta r_i^u)\}_{i=1}^m$; hence, $P_{ext}(x_j^c|r_c^u, \Omega_u)$ characterizes whether the sentence x_j^c about item c is qualified to characterize the desired opinion difference conditioned on user u's historical content Ω_u and target rating r_c^u. For example, a negative Δr_i^u suggests the opinion conveyed in x_j^c is expected to be less positive than that in x_i^u. On a similar note, $P_{ref}(x|x_j^c, r_c^u, \Omega_u)$ quantifies if x is a good rewriting of x_j^c to satisfy the desired opinion rating r_c^u for item c by user u.

One can parameterize $P_{ext}(x_j^c|r_c^u, \Omega_u)$ and $P_{ref}(x|x_j^c, r_c^u, \Omega_u)$ and estimate the corresponding parameters based on the maximum likelihood principle over observations in Ω_u. However, data likelihood alone is insufficient to generate high-quality explanations, as we should also emphasize on fluency, brevity, and diversity of the generated explanations. To realize this generalized objective, assume a metric $\pi(x|u, c)$ that measures the quality of generated explanation x for user u about item c, the training objective of CompExp is set to maximize the expected quality of its generated explanations under $\pi(x|u, c)$,

$$J = \mathbb{E}_{x \sim P(x|u,c,r_c^u)}[\pi(x|u, c)] \tag{2}$$

In this work, we present a customized BLUE score specifically for the comparative explanation generation problem to penalize short and generic content.

Next, we dive into the detailed design of CompExp in Section 3.1, then present our metric $\pi(x|u, c)$ for parameter estimation in Section 3.2 and 3.3, and finally illustrate how to estimate each component in CompExp end-to-end in Section 3.4.

3.1 Extract-and-Refine Architecture

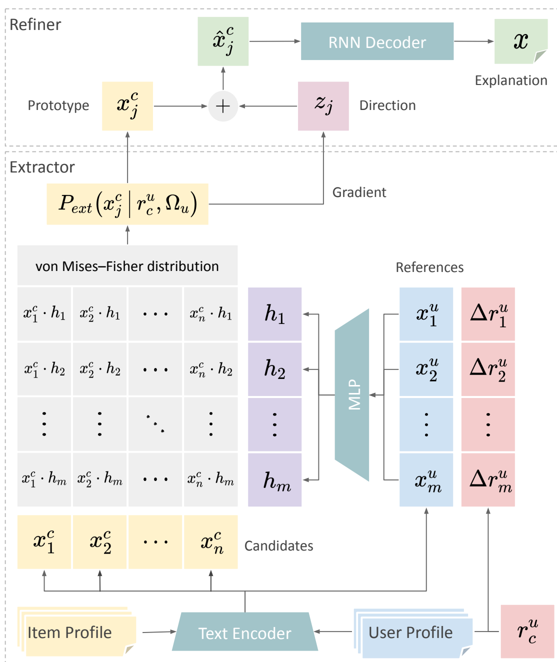

Figure 2: The extract-and-refine model architecture for CompExp. The extractor extracts a candidate sentence from item c's profile as a prototype for explanation generation; and the refiner rewrites this sentence to optimize the desired quality metric for comparative explanation.

Our proposed model architecture for CompExp is shown in Figure 2, which in a nutshell is a fully connected hierarchical neural network. The explanations for a user item pair (u, c) is generated via an extract-and-refine process, formally described in Eq Eq (1). Comparing to existing pure generation-based explanation methods (Li et al., 2017; Sun et al., 2020; Yang et al., 2021), one added benefit of our solution is to ensure faithfulness of the generated explanations: it avoids mentioning attributes that are not relevant to the target item. To address the limitations in directly using existing content, e.g., unaligned content style or sentiment polarity, the refinement step further rewrites the extracted sentence to make its content better fit for the purpose of comparative explanation, e.g., improve the quality defined by $\pi(x|u, c)$.

We refer to $P_{ext}(x_j^c|r_c^u, \Omega_u)$ as the extractor and $P_{ref}(x|x_j^c, r_c^u, \Omega_u)$ as the refiner. Next, we will zoom into each component to discuss its design principle and technical details.

• **Extractor.** The extractor's goal is to select a prototype sentence x_j^c from item c's profile Ψ_c for a given opinion rating r_c^u that best satisfies the comparativeness suggested by the user profile Ω_u. We refer to $x_j^c \in \Psi_c$ as an extraction candidate and $x_i^u \in \Omega_u$ as a reference. The extractor adopts a bidirectional GRU (Chung et al., 2014) as the universal text encoder to convert the extraction candidates and references into continuous embedding vectors. Since the pairwise comparison specified by Δr_i^u is a scalar, we use a one-hot vector to encode it when the ratings are discrete, otherwise we use a non-linear multi-layer perceptron (MLP) as the rating encoder.

Intuitively, in the one dimensional rating space, we can easily recover the intended sentence's rating r_c^u from the rating of the reference sentence r_i^u and required rating difference Δr_i^u. As an analogy, we consider the rating difference vector as the transform direction that suggests the ideal comparative explanation in the latent text space from a reference sentence x_i^u, denoted as $f(x_i^u, \Delta r_i^u) \rightarrow h_i$. As a result, h_i is the text embedding vector for the ideal comparative explanation. The extractor implements such a transformation using an MLP taking the concatenation of the text embedding and rating difference embedding vectors as input.

Given the desired comparative explanation h_i, the extraction candidates can be evaluated by their similarities towards h_i. This specifies a directional distribution $Q(x; h_i)$ centered on h_i in the latent text embedding space. Since cosine is a commonly used similarity metric for text embeddings, we formulate $Q(x; h_i)$ as a von Mises-Fisher distribution (Guu et al., 2018) over all the extraction candidates,

$$Q(x; h_i) \propto f_{vMF}(x; h_i, \kappa) = C_p(\kappa) e^{\kappa \cos(x, h_i)}$$

where $f_{vMF}(\cdot)$ is the probability density function, κ is the concentration parameter, and $C_p(\kappa)$ is a normalization function about k. Because each reference sentence x_i^u will suggest a different directional distribution, we extend the von Mises-Fisher distribution to cover multiple centriods and define $P_{ext}(x_j^c|r_c^u, \Omega_u)$ as follows,

$$P_{ext}(x_j^c|r_c^u, \Omega_u) \propto \sum_{x_i^u \in \Omega_u} f_{vMF}\Big(x_j^c; f(x_i^u, \Delta r_i^u), \kappa\Big) \tag{3}$$

Intuitively, in Eq Eq (3), each ideal embedding h_i suggests which extraction candidate better fits the comparativeness embedded in Ω_u. The summation over Ω_u aggregates each reference sentence's evaluation on candidate sentence x_j^c. κ is kept as a hyper-parameter which shapes the extraction probability distribution: a larger κ value leads to a skewer distribution. We can use it to control the exploration of the extraction candidates during the policy gradient based model training, which will be introduced in Section 3.4.

• **Refiner.** The objective of the refiner is to rewrite the extracted prototype to further improve the quality metric $\pi(x|u, c)$. As we argued before, a better explanation should be more supportive to the pairwise comparison required by the user profile. Therefore, assuming the refiner successfully turns the prototype x_j^c into a better framed sentence $\hat{x}_j^c$ about the item c for user u, then when we give $\hat{x}_j^c$ back to the extractor together with x_j^c, the extractor should prefer the revised version over the original one. Otherwise, we should keep refining $\hat{x}_j^c$ until the extractor believes it can no longer be improved. Hence, the refiner needs to find a direction such that $P_{ext}(x_j^c|r_c^u, \Omega_u) < P_{ext}(\hat{x}_j^c|r_c^u, \Omega_u)$, which is exactly suggested by the gradient of $P_{ext}(x_j^c|r_c^u, \Omega_u)$ with respect to x_j^c, i.e., the fastest direction for x_j^c to increase the value of $P_{ext}(x_j^c|r_c^u, \Omega_u)$. As a result, our refiner simply pushes the text embedding vector of x_j^c alone this gradient direction:

$$\begin{aligned} z_j &= \nabla_{x_j^c} P_{ext}(x_j^c | r_c^u, \Omega_u) \\ &\propto \sum_i^m e^{\kappa \cos(x_j^c, h_i)} \big[\frac{h_i}{|x_j^c||h_i|} - \cos(x_j^c, h_i) \frac{x_j^c}{|x_j^c|^2} \big] \end{aligned}$$

Since the refinement step should only polish the extracted prototype instead of dramatically changing it, we normalize the gradient to a unit vector and restrict the step size to one in all cases, i.e., $\hat{x}_j^c = x_j^c + z_j/|z_j|$. At last, we include a single-layer GRU with attention (Luong et al., 2015) as the text decoder to convert the refined text vector $\hat{x}_j^u$ to the final explanation sentence x.

Connecting these two modules together, CompExp generates explanations for a ranked list of recommended items one at a time. To understand why the generated explanations carry comparativeness, we can consider the user's profile Ω_u as an anchor. Because all the explanations are generated against this anchor, the comparisons among the explanations emerge.

3.2 Explanation Quality Metric

To train CompExp under Eq Eq (2), we need to define the explanation quality metric $\pi(x|u, c)$. There is no commonly agreed offline metric for explanation quality in the community yet. And obtaining real user feedback is not feasible for offline model training. Currently, most of explainable recommendation solutions (Sun et al., 2020; Li et al., 2017; Yang et al., 2021) adopt metrics measuring the overlapping content between the generated explanations and user reviews, such as BLEU (Papineni et al., 2002).

However, the BLEU metric, which is initially designed for machine translation, is problematic in explanation evaluation for at least two important reasons. First, it is biased towards shorter sentences. As a precision-based metric, BLEU overcomes the short-length issue by introducing the brevity penalty, which down-scales the precision when the generated length is smaller than its "best match length" (Papineni et al., 2002). The "best match length" design is reasonable in machine translation, because all reference sentences are valid translations covering the information contained in the source language, regardless of their length differences. However, when using review sentences as proxies of explanations, the reference sentences from one review can describe totally different aspects of the same item and vary significantly in length and information contained. Since short-length generation benefits precision (less prone to erroneous word choices), BLEU favors explanations exploiting the short references as the "best match". As a result, it pushes the models to generate explanations that are generally much shorter than the average sentence length in a review, and hence fails to explain the item in details. Second, though precision-based, BLEU is incapable to differentiate the importance of different words in a reference sentence. Words are valued equally in machine translation, but their impact in explanations varies significantly to users. For example, in Figure 1, the feature and descriptive words such as "swimming pool" and "friendly" help users better understand the target item than a very frequent but generic word, like "hotel" and "good". BLEU's indiscrimination to words unavoidably favors the explanations with more generic content due to their higher chance of appearance. We later demonstrate how the BLEU metric led to both short and generic explanations in our experiments.

To design a more appropriate metric to evaluate the explanation quality and better guide our model training, we propose IDF-BLEU, i.e., Inverse Document Frequency (IDF) enhanced BLEU. It introduces three changes on top of BLEU to balance the important factors in explanations: length, content overlapping, and content rarity.

First, to penalize an overly short generation, we replace the "best match length" in the brevity factor with the average length of sentences from all reviews,

$$BP_{len} = e^{\min(1 - \frac{l_r}{l_x}, 0)}$$

where l_r and l_x is the average length of references and the length of the explanation respectively. Second, to differentiate the importance of different words, we introduce IDF to measure the value of n-grams and use it to reweigh the precision in BLEU. We compute the IDF of word g by the number of sentences where it occurs,

$$IDF(g) = log\frac{S}{s_g} + 1$$

where S is the total number of review sentences in the training corpus and s_g is the number of sentences containing word g. We approximate the IDF of an n-gram by the largest IDF of its constituent words. Then the clipped n-gram precision in BLEU is modified as

$$p_n = \frac{\sum_{g^n \in x} IDF(g^n) \cdot Count_{clip}(g^n)}{\sum_{g^n \in x} IDF(g^n) \cdot Count(g^n)} \tag{4}$$

where g^n represents the n-gram and $Count_{clip}(g^n)$ is the BLEU's operation to calculate the count of g^n in sentence x while being clipped by the corresponding maximum count in the references. Through the reweighing, correctly predicting an informative word becomes more rewarding than a generic word. However, it alone cannot evaluate content rarity, since the precision-based metric cannot punish sentences for not including rare words. Therefore, at last, inspired by the length brevity factor in original BLEU, we introduce a similar IDF brevity factor to punish sentences lacking words with high IDF,

$$BP_{IDF} = e^{\min(1-\frac{d_r}{d_x},0)}$$

where d_x is the average IDF per word $d_x = \sum_{g \in x} IDF(g)/l_x$ and d_r is corresponding average value in references. Then combining them forms our IDF-BLEU,

$$IDF\text{-}BLEU = BP_{len} \cdot BP_{IDF} \cdot \exp\Big(\sum_{n=1}^{N} w_n \log p_n\Big) \tag{5}$$

where w_n is BLEU's parameter used as the weight of the n-gram precision. We use the proposed IDF-BLEU as the quality metric $\pi(x|u,c)$ for CompExp training.

3.3 Hierarchical Rewards

CompExp is a fully connected neural network which can be trained end-to-end with the gradient derived from Eq Eq (2). However, blind end-to-end training faces the risk that the model violates the purpose of our designed extract-and-refine procedure, as the model has a great degree of freedom to arbitrarily push the prototype x_j^c in the continuous vector space to optimize Eq Eq (2). For example, it could disregard the extracted prototype and generate totally irrelevant content to the target item c in the refiner.

To enforce the extract-and-refine workflow, we introduce additional intrinsic reward for each layer respectively to regularize their behaviours. Specifically, as IDF-BLEU is used to measure the explanation quality in Eq Eq (2), we directly use the extracted sentence's IDF-BLEU to reward the extractor, i.e., introduce $\pi_{ext}(x_j^c|u,c) = IDF\text{-}BLEU(x_j^c)$. For the refiner, we discourage it in pushing the final generation too far away from the extracted one. Inspired by the clipped precision in Eq Eq (4), we propose a clipped recall to measure how many words from the selected sentence x_j^c are still covered in the refined sentence,

$$a_n = \frac{\sum_{g^n \in x_j^c} IDF(g^n) \cdot min[Count_{clip}(g^n), Count_x(g^n)]}{\sum_{g^n \in x_j^c} IDF(g^n) \cdot Count_{clip}(g^n)} \tag{6}$$

where $Count_{clip}(g^n)$ is the clipped count of n-gram g^n towards the references like in BLEU, and $Count_x(g^n)$ is the count of g^n in the refined explanation x. In other words, the denominator is the prototype's overlap with the target references and the numerator is the overlap among the prototype, references, and the final explanation. We did not use classical recall definition because it would reward the refiner to retain the entire prototype. We only encourage the refiner to keep the n-grams that are actually presented in the references. We compute the refiner's intrinsic reward by aggregating the clipped recall over different n-grams $\pi_{ref}(x, x_j^c) = \exp\left(\sum_{n=1}^{N} w_n \log a_n\right)$. We did not provide this reward to the extractor, because it biases the extractor to short and generic candidates which are easier for the refiner to cover.

With the hierarchical intrinsic rewards introduced for each component, we can optimize Eq Eq (2) by policy gradient as

$$\begin{aligned}\nabla_\Theta J \approx &[\lambda_1 \pi(x|u,c) + \lambda_2 \pi_{ref}(x, x_j^c)] \nabla_\Theta \log P_{ref}(x|x_j^c, r_c^u, \Omega_u) \\ &+ [\lambda_3 \pi(x|u,c) + \lambda_4 \pi_{ext}(x_j^c)] \nabla_\Theta \log P_{ext}(x_j^c|r_c^u, \Omega_u)\end{aligned}$$

where λ_1 to λ_4 are coefficients to adjust the importance of each reward, and Θ stands for the model parameters in CompExp.

3.4 Model Training

The whole model training process can be organized into two steps: pre-training and fine-tuning. The pre-training step aims to bootstrap the extractor and refiner independently. To prepare the extractor to recognize the comparative relationships among sentences, we treat every observed review sentence as the extraction target and train the extractor to maximize its negative log-likelihood with regard to the corresponding user and item profiles.

It is important to pre-train the refiner as a generative language model, because it would be very inefficient to learn all the natural language model parameters only through the end-to-end training. However, we do not have any paired sentences to pre-train the refiner. We borrowed the method introduced in (Guu et al., 2018; Weston et al., 2018) to manually craft such pairs. Specifically, for every sentence, we compute its cosine similarity against all other sentences in the same item profile in the latent embedding space, and select the most similar one to pair with. Then we use this dataset to pre-train the refiner with negative log-likelihood loss.

In the fine-tuning stage, we concatenate the pre-trained layers and conduct the end-to-end training with policy gradient. To make the policy gradient training more resilient to variance and converge faster, it is important to have a baseline to update the model with reward advantages instead of using the rewards directly. We apply Monte Carlo sampling in both extractor and refiner to have multiple explanations, and use their mean rewards as the baseline.

4. Experimental Evaluations

We demonstrate empirically that our proposed solution CompExp can generate improved explanations compared to state-of-the-art explainable recommendation algorithms, especially about the explanations' comparativeness. We conduct experiments on two different recommendation scenarios: RateBeer reviews with single-ratings (McAuley et al., 2012) and TripAdvisor reviews with *multi-aspect ratings* (Wang et al., 2010).

4.1 Experiment Setup

As our solution only focuses on explanation generation, it can be applied to any recommendation algorithm of choice. In our experiments, we directly use the ground-truth review ratings as the recommendation score to factor out any deviation or noise introduced by specific recommendation algorithms. For completeness, we also empirically studied the impact from input ratings if switched to a real recommendation algorithm's predictions.

Table 1: Summary of the processed datasets.

Dataset	# Users	# Items	# Reviews	Rating Range
RateBeer	6,566	19,876	2,236,278	0 - 20
TripAdvisor	4,954	4,493	287,879	1 - 5

4.1.1 Data Pre-Processing

In the RateBeer dataset, we segment each review into sentences, and label them with the overall ratings from their original reviews. In the TripAdvisor dataset, there are separate ratings for five aspects including *service, room, location, value and cleanliness*. Therefore, each TripAdvisor review is expected to be a mix of a user's opinions on these different aspects about the item. We segment sentences in a TripAdvisor review to different aspects using the boot-strapping method from (Wang et al., 2010) and assign resulting sentences the corresponding aspect ratings. These two datasets evaluate CompExp under different scenarios: overall opinion vs., aspect-specific opinion in users' opinionated content.

We also adopt the recursive filtering method introduced in (Wang et al., 2018a) to alleviate the common sparsity issue in the review datasets. The statistics of the processed datasets are summarized in Table 1. We build the vocabulary of each dataset by selecting the 20,000 most frequent words and mapping others to unknown. Each dataset is split into 70% for training, 15% for validation, and 15% for testing.

4.1.2 Baselines

We compared with three explainable recommendation baselines that generate natural language explanations, covering both extraction-based and generation-based solutions.

- **NARRE**: Neural Attentional Regression model with Review-level Explanations (Chen et al., 2018). It is an extraction-based solution. It learns the usefulness of the existing reviews through attention and selects the most attentive reviews as the explanation.

- **NRT**: Neural Rating and Tips Generation (Li et al., 2017). It is a generation-based solution. It models rating regression and content generation as a multi-task learning problem with shared user-item latent space. Explanations are directly generated from its neural language model component.

- **SAER**: Sentiment Aligned Explainable Recommendation (Yang et al., 2021). This is another generation-based solution using multi-task learning to model rating regression and explanation generation. But it focuses specifically on the sentiment alignment between the predicted rating and generated explanation.

We include three variants of CompExp to better demonstrate the effect of each component in it:

- **CompExp-Ext**: the extractor of our solution. It directly uses the selected sentences as explanations without any refinement. This variant helps us study how the extractor works and also serves as a fair counterpart for the other extraction-based baselines.

- **CompExp-Pretrain**: our model with pre-training only, which is a simple concatenation of the separately trained extractor and refiner without joint training. We compare it with CompExp to show the importance of end-to-end policy gradient training.

- **CompExp-BLEU**: our model trained with BLEU instead of IDF-BLEU. We create this variant to demonstrate the flaws of using BLEU to evaluate the quality of generated explanations.

Table 2: Explanation quality evaluated under IDF-BLEU, BLEU, average sentence length, average IDF per word, rep/l, seq_rep_2, feature precision and recall on RateBeer and TripAdvisor datasets.

Model	IDF-BLEU 1	IDF-BLEU 2	IDF-BLEU 4	BLEU 1	BLEU 2	BLEU 4	Avg Length	IDF/word	rep/l	seq_rep_2	Feature precision	Feature recall
RateBeer												
Human	/	/	/	/	/	/	11.13	2.45	0.0535	0.0015	/	/
NARRE	17.00	5.18	1.29	30.22	9.90	3.58	11.50	2.43	0.0643	0.0013	0.2217	0.0722
NRT	30.38	16.30	5.80	48.22	25.28	10.03	10.43	2.09	0.1123	0.0240	0.4563	0.1320
SAER	31.79	16.02	5.71	49.08	26.87	10.59	10.71	1.93	0.1146	0.0223	0.4751	0.1347
CompExp-Ext	24.86	11.72	2.99	38.54	18.74	5.98	12.10	2.36	0.0420	0.0010	0.3092	0.0929
CompExp-Pretrain	27.59	13.44	4.19	44.93	21.53	7.95	10.55	2.07	0.1448	0.0381	0.3922	0.1123
CompExp-BLEU	23.20	14.55	4.70	**53.45**	**32.42**	**11.62**	7.09	1.83	0.0266	0.0006	0.4025	0.1173
CompExp	**32.36**	**19.55**	**6.95**	49.14	29.63	11.41	10.52	2.16	0.0572	0.0057	**0.4796**	**0.1383**
TripAdvisor												
Human	/	/	/	/	/	/	12.85	2.45	0.0604	0.0021	/	/
NARRE	11.97	3.43	1.59	20.45	6.23	3.38	13.17	2.41	0.0641	0.0022	0.1733	0.1258
NRT	16.19	7.50	**2.48**	30.62	13.07	5.11	10.22	1.81	0.1277	0.0135	0.2939	0.1866
SAER	16.37	7.65	2.35	31.20	13.51	4.94	10.08	1.71	0.1361	0.0141	**0.3178**	**0.1961**
CompExp-Ext	13.52	4.25	1.14	22.12	7.30	2.66	14.70	2.39	0.0726	0.0037	0.2218	0.1553
CompExp-Pretrain	14.50	6.11	1.99	27.14	11.12	4.32	10.79	1.92	0.1177	0.0250	0.2736	0.1597
CompExp-BLEU	17.04	7.39	2.04	**32.67**	**14.66**	**5.53**	10.77	1.76	0.1597	0.0277	0.2332	0.1637
CompExp	**21.35**	**8.01**	2.16	31.70	12.23	4.16	13.35	2.12	0.0654	0.0053	0.3155	0.1930

4.2 Quality of Generated Explanations

To comprehensively study the quality of generated explanations, we employ different types of performance metrics, including IDF-BLEU-{1, 2, 4}, BLEU-{1, 2, 4}, average sentence length, average IDF per word, rep/l and seq_rep_2, and feature precision & recall. Both rep/l and seq_rep_2 are proposed in (Welleck et al., 2019) to evaluate content repetition and higher values mean the content is more repetitive. Features are items' representative attributes that users usually care the most (Wang et al., 2018a; Xian et al., 2021; Yang et al., 2021), e.g., "pool" in Figure 1. The precision and recall measure if features mentioned in the generated explanations also appear in the user's ground-truth review. We also include ground-truth review sentences as a reference baseline (labeled as "Human") to study the differences between human and algorithm generated content. The results are reported in Table 2.

While CompExp-BLEU topped every BLEU category on both datasets, CompExp also led under almost all IDF-BLEU categories except IDF-BLEU-4 on TripAdvisor. This shows the effectiveness of our model design and the importance of directly optimizing the target evaluation metrics. To understand whether IDF-BLEU is a better metric than BLEU in evaluating the generated explanations, we should consider how the "ground-truth" content from real users look like, e.g., their average length and IDF/word, which suggest how much information is usually contained in a user-written sentence. As we can clearly notice that Avg Length and IDF/word in CompExp-BLEU are much smaller than Human. This suggests simply optimizing BLEU led to much shorter and less informative content. This follows our discussion before: BLEU encourage a model to generate less words and abuse common words to achieve high n-gram precision. CompExp-BLEU's low feature precision and recall also reflect its weakness in providing informative content. Therefore, the witnessed "advantages" of CompExp-BLEU in BLEU most likely come from shorter and more generic sentences, instead of really being closer to the ground-truth content.

There is clear performance gap between the extraction-based solutions (NARRE, CompExp-Ext) and generation-based ones (NRT, SAER, CompExp). While generation-based solutions largely outperformed extraction-based ones in content overlapping with ground-truth (IDF-BLEU, BLEU, feature precision and recall), they were generally very different from human writings in terms of sentence length, use of rare words (IDF/word), and content repetition (rep/l, seq_res_2). The extraction-based solutions use content provided by human, but they are limited to the existing

Table 3: Case study of the explanations generated by different models.

Model	Sample 1	Sample 2
Human	aroma of caramel, cherry, raisins, and florals.	pours clear yellow body with a small white head.
NARRE	the finish is dry and ashy.	not bad, if one is looking for a refreshing, light wheat beer.
NRT	flavor of chocolate, roasted malt, and light smoke.	the beer is a hazy yellow-orange color.
SAER	aroma of caramel, caramel, and citrus.	medium body, watery texture, and carbonation.
CompExp-Ext	sweet aroma with toasted malt, caramel and alcohol notes.	pours a hazy golden with a small white head.
CompExp	aroma of caramel, malt, and alcohol.	pours a hazy yellow body with a small white head.

content. The generation-based solutions customize content for each recommendation, but suffer from common flaws of generative models, e.g., short, dull, and repetitive. Among all the models, CompExp achieved the best balance among all metrics. It significantly exceeded all baselines in terms of IDF-BLEU-{1,2} and its BLEU was only behind CompExp-BLEU. Its feature precision and recall are competitive with SAER while leading the rest, though SAER enjoys additional advantage from predefined feature pool of each item as input. As a generation-based model, CompExp largely improved the average length, word rarity, and reduced repetition over NRT and SAER. The only exception is that CompExp-BLEU was less repetitive in RateBeer, but it is mainly because its explanations were very short in general.

Although CompExp performed well as a whole, it is inspiring to investigate if each component works as expected in the extract-and-refine workflow. First, both CompExp-Ext and NARRE are extraction-based and share the same candidate pool, but CompExp-Ext showed obvious advantage under most categories of these two metrics. It suggests our extractor alone can act as a competent solution where generation-based models do not fit, e.g., critical real-time applications requiring minimum response time. The comparison between CompExp-Ext and CompExp-Pretrain demonstrates that the refiner is able to leverage the gradient direction to improve the prototypes, even when the prototypes are given by an extractor that has not been trained jointly with the refiner. At last, there are huge gaps in all metrics between CompExp-Pretrain and CompExp in both datasets. It is obvious that our reward design is beneficial to both quality and diversity of the generated explanations.

Case Study. Groups of example explanations generated by CompExp and other baselines are shown in Table 3. The ground-truth explanations are given for reference denoted as *Human*. Comparing NARRE and CompExp-Ext shows the value of modeling comparativeness in users provided historical content. Sentences extracted by CompExp-Ext are much closer to the ground-truth than NARRE's. Comparing CompExp-Ext and CompExp shows the effectiveness our rewriting module in improving the explanation quality. For example, in Sample 1, the extracted explanation correctly covers the attribute "aroma" and "caramel", but its sentence structure is different from the ground-truth's. The refined explanation keeps the two correct attributes and improves the sentence structure. In Sample 2, while the extractor picks a sentence almost the same as the ground-truth, the refiner further changes the word "golden" to "yellow", which better reflects the user's preference in wording.

Comparativeness. To verify if the generated explanations by CompExp capture the comparative ranking of items, we study its its output's sensitivity to the input recommendation ratings. As a starting point, the ground-truth explanation perfectly aligns with the recommendation ranking, which is derived from the ground-truth rating. If the generated explanation carries the same ranking of item, the generated content should be close to the ground-truth content. As a result, if we manipulate the input recommendation scores of items, the generated explanations should start to deviate. The further we push the rankings apart, the further the generated explanation should be pushed away from the ground-truth explanation. We use IDF-BLEU and BLEU to measure the content similarity and perturb the recommendation ratings with Gaussian noise. As shown in Figure 3a, all IDF-BLEU and BLEU metrics keep decreasing with the increasing amount of perturbation. In other words, even if it is for the same user and same set of items, with different recommendation scores assigned, CompExp would generate different explanations to explain their relative ranking.

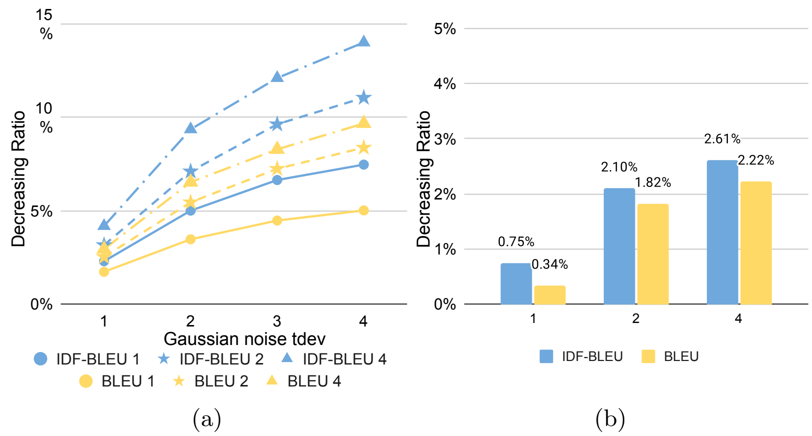

Figure 3: (a) Impact of noise in recommendation ratings on BLEU and IDF-BLEU. (b) Change in BLEU and IDF-BLEU with algorithm's predicted ratings.

Predicted Ratings. Motivated by the findings in Figure 3a, we further study how CompExp is influenced by a real recommendation algorithm's predicted ratings. We employed the neural collaborative filtering algorithm (He et al., 2017) and used its predicted ratings in CompExp's training and testing. The result is plotted in Figure 3b. Compared with previous randomly perturbed ratings, the predicted ratings bring very limited changes to the explanations. This confirms our experiment results based on ground-truth ratings can fairly represent CompExp's performance in real-world usage scenarios.

5. User Study

We have three research questions to be answered by user study: 1) does users' judgement toward explanation quality aligns more with IDF-BLEU than with BLEU; 2) do users find our comparative explanations more helpful than the baselines'; and 3) can users better perceive the comparative ranking from our explanations than the baselines'. To answer these three research questions, we design two user study tasks based on RateBeer dataset using Amazon Mechanical Turk.

The first task studies the first two research questions together. Specifically, we shuffle explanations from different models about the same recommended item and ask the participants to compare them, and then select the most helpful ones. To help participants evaluate the explanation quality, we include the original user review as the item description, towards which they can judge if the content in the explanations is accurate or informative. For each recommended item, we ask participants to answer the following question after reading its description and the candidate explanations:

> *"Which of the following explanations best describe the characteristics of the given beer and help you the most to understand why you should pay attention to the recommendation?"*

Participants are allowed to choose multiple explanations if they feel the choices are equally good. In this experiment, we compared explanations from CompExp, SAER, NRT, and NARRE; and in the end, we collected 660 user responses.

The results are presented in Table 4 and 5. In Table 4, we used Cohen's kappa coefficient to compare IDF-BLEU and BLEU's agreement with users' responses. For each test case, we pair explanations that the participants chose as helpful with the rest to form a set of explanation pairs. Then we use IDF-BLEU-{1,2,4} and BLEU-{1,2,4} to identify the helpful one in each pair. The kappa coefficient shows that IDF-BLEU aligns significantly better with users' judgment in all three subcategories under paired t-test. Table 5 shows the helpfulness vote on each model and the paired

Table 4: Cohen's kappa coefficient of explanation quality between the human judgements and BLEU & IDF-BLEU.

		1	2	4
κ	BLEU	0.2936	0.3114	0.2814
	IDF-BLEU	0.3452	0.3396	0.3152
Paired t-test		0.0001	0.0094	0.0071

Table 5: Up-vote rate of explanations' helpfulness.

	CompExp	SAER	NRT	NARRE
Up-vote Rate	43.79%	37.27%	35.61%	30.61%
Paired t-test (CompExp vs .)	/	0.0182	0.009	0

t-test results of CompExp against other baselines. The helpfulness vote on CompExp is significantly higher than others, which suggests strong user preference over its generated explanations.

The second task focuses on the last research question, i.e., if a user is able to perceive the ranking of recommended items from the provided explanations. In this task, we randomly paired items of different ratings and asked participants to identify which item is better by reading the provided explanations. We then evaluated the agreement rate between participants' choices and the actual ranking. In particular, we presented explanations from different models and required the participants to answer the following question:

> *"After reading the explanations for recommended items, which item would you like to choose? You are expected to judge the quality of the recommended items based on the provided explanations."*

We chose SAER and NRT as baselines. Besides, we also include the ground-truth sentences from the actual user reviews as a reference. We collected 200 responses for each model.

Table 6 reports the agreement rates between the actual ranking and the ranking perceived by the participants. CompExp's agreement rate is higher than NRT and slightly higher than SAER, but it is far below the Ground-Truth. The Ground-Truth's high agreement rate quantitatively confirms that the original user provided review sentences are highly comparative. This observation supports our choice of training the comparative explanation generation from paired user review sentences. And it also suggests there is still a performance gap for learning-based solutions to bridge in achieving the level of comparativeness in the generated explanations. And an improved metric for optimization, e.g., introduce pairwise comparativeness into IDF-BLEU might be a promising direction.

6. Conclusions and future work

In this paper, we studied the problem of comparative explanation generation in explainable recommendation. The objective of our generated explanations is to help users understand the comparative item rankings provided in a recommender system. We develop a neural extract-and-refine architecture to generate such comparative explanations, with customized metrics to penalize generic and useless content in the generated explanations. Both offline evaluations and user studies demonstrated the effectiveness of our solution.

This work starts a bright new direction in explainable recommendation. Our current solution only focuses on explanation generation, by assuming a perfect recommendation algorithm (i.e., we directly used the ground-truth opinion ratings in our experiments). It is important to improve our model by co-design with a real recommendation algorithm, whose recommendation score is expected

Table 6: Agreement rate between actual ranking and the users perceived ranking of paired items based on the provided explanations.

	GT	CompExp	SAER	NRT
Agreement Rate	72.29%	57.27%	56.25%	53.14%

to be noise and erroneous. In addition, we still heavily depend on existing review content to guide explanation generation. It will be more meaningful to introduce actual user feedback in this process, i.e., interactive optimization of explanation generation.

Acknowledgments

This work is partially supported by the National Science Foundation under grant SCH-1838615, IIS-1553568, and IIS-2007492, and by Alibaba Group through Alibaba Innovative Research Program.

References

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

Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms*, 11(9):137, 2018.

Marko Balabanović and Yoav Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

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

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.

Hongxu Chen, Yicong Li, Xiangguo Sun, Guandong Xu, and Hongzhi Yin. Temporal meta-path guided explainable recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 1056–1064, 2021.

Tong Chen, Hongzhi Yin, Guanhua Ye, Zi Huang, Yang Wang, and Meng Wang. Try this instead: Personalized and interpretable substitute recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 891–900, 2020.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics*, 6:437–450, 2018.

Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1661–1670, 2015.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

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.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Ke Ji and Hong Shen. Jointly modeling content, social network and ratings for explainable and cold-start recommendation. *Neurocomputing*, 218:1–12, 2016.

Alexandros Karatzoglou, Linas Baltrunas, and Yue Shi. Learning to rank for recommender systems. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pages 493–494, 2013.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

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.

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.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774, 2017.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

Julian McAuley, Jure Leskovec, and Dan Jurafsky. Learning attitudes and attributes from multi-aspect reviews. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining*, ICDM ’12, page 1020–1025, USA, 2012. IEEE Computer Society. ISBN 9780769549057.

Julian McAuley, Rahul Pandey, and Jure Leskovec. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 785–794, 2015.

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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

Steffen Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000. IEEE, 2010.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

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.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295, 2001.

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.

Peijie Sun, Le Wu, Kun Zhang, Yanjie Fu, Richang Hong, and Meng Wang. Dual learning for explainable recommendation: Towards unifying user preference prediction and review generation. In *Proceedings of The Web Conference 2020*, pages 837–847, 2020.

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.

Quoc-Tuan Truong and Hady Lauw. Multimodal review generation for recommender systems. In *The World Wide Web Conference*, pages 1864–1874, 2019.

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.

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, 2018a.

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, 2018b.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*, 2019.

Jason Weston, Emily Dinan, and Alexander H Miller. Retrieve and refine: Improved sequence generation models for dialogue. *arXiv preprint arXiv:1808.04776*, 2018.

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.

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.

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.

Yongfeng Zhang and Xu Chen. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval*, 14(1):1–101, 2020.

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.