Interacting with Explanations through Critiquing

Diego Antognini[1] , **Claudiu Musat**[2] and **Boi Faltings**[1]
[1]École Polytechnique Fédérale de Lausanne, Switzerland
[2]Swisscom, Switzerland
{diego.antognini, boi.faltings}@epfl.ch, claudiu.musat@swisscom.com

Abstract

Using personalized explanations to support recommendations has been shown to increase trust and perceived quality. However, to actually obtain better recommendations, there needs to be a means for users to modify the recommendation criteria by interacting with the explanation. We present a novel explanation technique using aspect markers that learns to generate personalized explanations of recommendations from review texts, and we show that human users significantly prefer these explanations over those produced by state-of-the-art techniques.

Our work's most important innovation is that it allows users to react to a recommendation by critiquing the textual explanation: removing (symmetrically adding) certain aspects they dislike or that are no longer relevant (symmetrically that are of interest). The system updates its user model and the resulting recommendations according to the critique. This is based on a novel unsupervised critiquing method for single- and multi-step critiquing with textual explanations. Empirical results show that our system achieves good performance in adapting to the preferences expressed in multi-step critiquing and generates consistent explanations.

1 Introduction

Explanations of recommendations are beneficial. Modern recommender systems accurately capture users' preferences and achieve high performance. But, their performance comes at the cost of increased complexity, which makes them seem like black boxes to users. This may result in distrust or rejection of the recommendations [Tintarev and Masthoff, 2015].

There is thus value in providing *textual explanations* of the recommendations, especially on e-commerce websites, because such explanations enable users to understand why a particular item has been suggested and hence to make better decisions [Kunkel *et al.*, 2018]. Furthermore, explanations increase overall system transparency [Tintarev and Masthoff, 2015] and trustworthiness [Zhang and Curley, 2018].

However, not all explanations are equivalent. [Kunkel *et al.*, 2019] showed that highly personalized justifications using

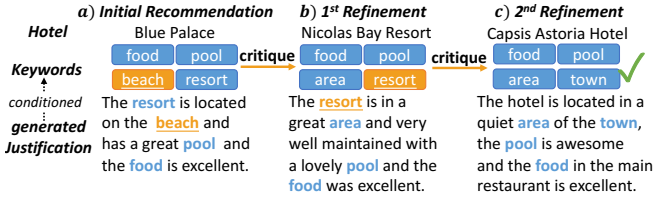


Figure 1: A flow of conversational critiquing over two time steps. a) The system proposes to the user a recommendation with a keyphrase explanation and a justification. The user can interact with the explanation and critique phrases. b) A new recommendation is produced from the user's profile and the critique. 3) This process repeats until the user accepts the recommendation and ceases to provide critiques.

natural language lead to substantial increases in perceived recommendation quality and trustworthiness compared to simpler explanations, such as aspect, template, or similarity.

A second, and more important, benefit of explanations is that they provide a basis for feedback: if a user is unsatisfied with a recommendation, understanding what generated it allows them to *critique* it (Fig. 1). Critiquing – a conversational method of incorporating user preference feedback regarding item attributes into the recommended list of items – has several advantages. First, it allows the system to correct and improve an incomplete or inaccurate model of the user's preferences, which improves the user's decision accuracy [Chen and Pu, 2012]. Compared to preference elicitation, critiquing is more flexible: users can express preferences in any order and on any criteria [Reilly *et al.*, 2005].

Useful explanations are hard to generate. Prior research has employed users' reviews to capture their preferences and writing styles (e.g., [Dong *et al.*, 2017]). From past reviews, they generate *synthetic* ones that serve as personalized *explanations* of ratings given by users. However, many reviews are noisy, because they partly describe experiences or endorsements. It is thus nontrivial to identify meaningful justifications inside reviews. [Ni *et al.*, 2019] proposed a pipeline for identifying justifications from reviews and asked humans to annotate them. [Chen *et al.*, 2019; Chen *et al.*, 2020] set the justification as the first sentence. However, these notions of justification were ambiguous, and they assumed that a review contains only one justification.

Recently, [Antognini *et al.*, 2021] solved these shortcomings by introducing a justification extraction system with no

prior limits imposed on their number or structure. This is important because a user typically justifies his overall rating with multiple explanations: one for each aspect the user cares about [Musat and Faltings, 2015]. The authors showed that there is a connection between faceted ratings and snippets within the reviews: for each subrating, there exists at least one text fragment that alone suffices to make the prediction. They employed a sophisticated attention mechanism to favor long, meaningful word sequences; we call these ***markers***. Building upon their study, we show that these *markers* serve to create better user and item profiles and can inform better user-item pair justifications. Fig. 2 illustrates the pipeline.

From explanations to critiquing. To reflect the overlap between the profiles of a user and an item, one can produce a set of keyphrases and then a synthetic justification. The user can correct his profile, captured by the system, by *critiquing* certain aspects he does not like or that are missing or not relevant anymore and obtain a new justification (Fig. 1). [Wu *et al.*, 2019] introduced a keyphrase-based critiquing method in which attributes are mined from reviews, and users interact with them. However, their models need an extra autoencoder to project the critique back into the latent space, and it is unclear how the models behave in multi-step critiquing.

We overcome these drawbacks by casting the critiquing as an unsupervised attribute transfer task: altering a keyphrase explanation of a user-item pair representation to the critique. To this end, we entangle the user-item pair with the explanation in the same latent space. At inference, the keyphrase classifier modulates the latent representation until the classifier identifies it as the critique vector.

In this work, we address the problem recommendation with fine-grained explanations. We first demonstrate how to extract multiple relevant and personalized justifications from the user's reviews to build a profile that reflects his preferences and writing style (Fig. 2). Second, we propose T-RECS, a recommender with explanations. T-RECS explains a rating by first inferring a set of keyphrases describing the intersection between the profiles of a user and an item. Conditioned on the keyphrases, the model generates a synthetic personalized justification. We then leverage these explanations in an unsupervised critiquing method for single- and multi-step critiquing. We evaluate our model using two real-world recommendation datasets. T-RECS outperforms strong baselines in explanation generation, effectively re-ranks recommended items in single-step critiquing. Finally, T-RECS also better models the user's preferences in multi-step critiquing while generating consistent textual justifications.

2 Related Work

2.1 Textual Explainable Recommendation

Researchers have investigated many approaches to generating textual explanations of recommended items for users. [McAuley and Leskovec, 2013] proposed a topic model to discover latent factors from reviews and explain recommendations. [Zhang *et al.*, 2014] improved the understandability of topic words and aspects by filling template sentences.

Another line of research has generated synthetic reviews as explanations. Prior studies have employed users' reviews and

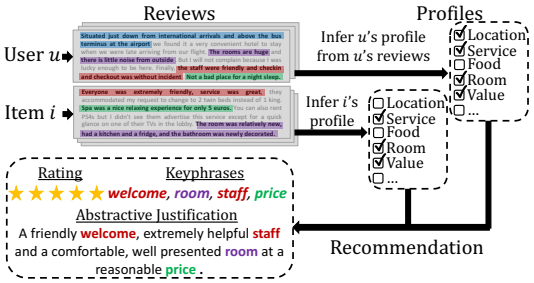


Figure 2: For reviews written by a user u and a set of reviews about an item i, we extract the justifications for each aspect rating and implicitly build an interest profile. T-RECS outputs a personalized recommendation with two explanations: the keyphrases reflecting the overlap between the two profiles, and a synthetic justification conditioned on the latter.

tips to capture their preferences and writing styles. [Catherine and Cohen, 2017] predicted and explained ratings by encoding the user's review and identifying similar reviews. [Chen *et al.*, 2019] extended the previous work to generate short synthetic reviews. [Sun *et al.*, 2020] optimized both tasks in dual forms. [Dong *et al.*, 2017] proposed an attribute-to-sequence model to learn how to generate reviews given categorical attributes. [Ni and McAuley, 2018] improved review generation by leveraging aspect information using a seq-to-seq model with attention. Instead of reviews, others have generated tips [Li *et al.*, 2017; Li *et al.*, 2019]. However, the tips are scarce and uninformative [Chen *et al.*, 2019]; many reviews are noisy because they describe partially general experiences or endorsements [Ni *et al.*, 2019].

[Ni *et al.*, 2019] built a seq-to-seq model conditioned on the aspects to generate relevant explanations for an existing recommender system; the fine-grained aspects are provided by the user in the inference. They identified justifications from reviews by segmenting them into elementary discourse units (EDU) [Mann and Thompson, 1988] and asking annotators to label them as "good" or "bad" justifications. [Chen *et al.*, 2019] set the justification as the first sentence. All assumed that a review contains only one justification. Whereas their notions of justification were ambiguous, we extract multiple justifications from reviews using *markers* that justify subratings. Unlike their models, ours predicts keyphrases on which the justifications are conditioned and integrates critiquing.

2.2 Critiquing

Refining recommended items allows users to interact with the system until they are satisfied. Some methods are example critiquing [Williams and Tou, 1982], in which users critique a set of items; unit critiquing [Burke *et al.*, 1996], in which users critique an item's attribute and request another one instead; and compound critiquing [Reilly *et al.*, 2005] for more aspects. The major drawback of these approaches is the assumption of a fixed set of known attributes.

[Wu *et al.*, 2019] circumvented this limitation by extending the neural collaborative filtering model [He *et al.*, 2017]. First, the model explains a recommendation by predicting a set of keywords (mined from users' reviews). In [Chen *et al.*, 2020], based on [Chen *et al.*, 2019], the model samples only

Situated just down from international arrivals and above the bus terminus at the airport we found it a very convenient hotel to stay when we were late arriving from our flight and subsequently to catch our flight . **The rooms are clean** and there is little noise from outside . They rooms are not plush, but sufficient (there's the Intercontinental if you want more) The staff were friendly and checkin and checkout was without incident . They even held our rooms on request even though hotel policy is to let them go if unpaid post 16:00 (because you pay on checkin here). Not a bad place for a nights sleep .

Figure 3: Extracted justifications from a hotel review. The inferred *markers* depict the excerpts that explain the ratings of the aspects: Service, Cleanliness, Value, Room, and Location. We denote in **bold** the EDU-based justification from the model of [Ni *et al.*, 2019].

one keyword via the Gumbel-Softmax function. Our work applies a deterministic strategy similar to [Wu *et al.*, 2019].

Second, [Wu *et al.*, 2019] project the critiqued keyphrase explanations back into the latent space, via an autoencoder that perturbs the training, from which the rating and the explanation are predicted. In this manner, the user's critique modulates his latent representation. The model of [Chen *et al.*, 2020] is trained in a two-stage manner: one to perform recommendation and predict one keyword and another to learn critiquing from online feedback, which requires additional data. By contrast, our model is simpler and learns critiquing in an unsupervised fashion: it iteratively edits the latent representation until the new explanation matches the critique. Finally, [Luo *et al.*, 2020] examined various linear aggregation methods on latent representations for multi-step critiquing. In comparison, our gradient-based critiquing iteratively updates the latent representation for each critique.

3 Extracting Justifications from Reviews

In this section, we introduce the pipeline for extracting high-quality and personalized justifications from users' reviews. We claim that a user justifies his overall experience with multiple explanations: one for each aspect he cares about. Indeed, it has been shown that users write opinions about the topics they care about [Zhang *et al.*, 2014]. Thus, the pipeline must satisfy two requirements: 1. extract text snippets that reflect a rating or subrating, and 2. be data driven and scalable to mine massive review corpora and to construct a large personalized recommendation justification dataset.

[Antognini *et al.*, 2021] proposed the multi-target masker (MTM) to find text fragments that explain faceted ratings in an unsupervised manner. MTM fulfills the two requirements. For each word, the model computes a distribution over the aspect set, which corresponds to the aspect ratings (e.g., service, location) and "not aspect." In parallel, the model minimizes the number of selected words and discourages aspect transition between consecutive words. These two constraints guide the model to produce long, meaningful sequences of words called ***markers***. The model updates its parameters by using the inferred *markers* to predict the aspect sentiments jointly and improves the quality of the *markers* until convergence.

Given a review, MTM extracts the *markers* of each aspect. A sample is shown in Fig. 3. Similarly to [Ni *et al.*, 2019], we filter out *markers* that are unlikely to be suitable justifications: including third-person pronouns or being too short. We use the constituency parse tree to select *markers* are verb phrases.

4 T-RECS: A Multi-Task Transformer with Explanations and Critiquing

Fig. 4 depicts the pipeline and our proposed T-RECS model. Let U and I be the user and item sets. For each user $u \in U$ (respectively an item $i \in I$), we extract *markers* from the user's reviews on the training set, randomly select N_{just}, and build a justification reference J^u (symmetrically J^i).

Given a user u, an item i, and their justification history J^u and J^i, our goal is to predict 1. a rating $\boldsymbol{y}_r$, 2. a keyphrase explanation $\boldsymbol{y}_{kp}$ describing the relationship between u and i, and 3. a natural language justification $\boldsymbol{y}_{just} = \{w_1, ..., w_N\}$, where N is the length of the justification. $\boldsymbol{y}_{just}$ explains the rating $\boldsymbol{y}_r$ conditioned on $\boldsymbol{y}_{kp}$.

4.1 Model Overview

For each user and item, we extract *markers* from their past reviews (in the train set) and build their justification history J^u and J^i, respectively (see Section 3). T-RECS is divided into four submodels: an **Encoder** E, which produces the latent representation $\boldsymbol{z}$ from the historical justifications and latent factors of the user u and the item i; a **Rating Classifier** C^r, which classifies the rating $\hat{\boldsymbol{y}}_r$; a **Keyphrase Explainer** C^{kp}, which predicts the keyphrase explanation $\hat{\boldsymbol{y}}_{kp}$ of the latent representation $\boldsymbol{z}$; and a **Decoder** D, which decodes the justification $\hat{\boldsymbol{y}}_{just}$ from $\boldsymbol{z}$ conditioned on $\hat{\boldsymbol{y}}_{kp}$, encoded via the **Aspect Encoder** A. T-RECS involves four functions: $\boldsymbol{z} = E(u, i)$;$\hat{\boldsymbol{y}}_r = C^r(\boldsymbol{z})$;$\hat{\boldsymbol{y}}_{kp} = C^{kp}(\boldsymbol{z})$; $\hat{\boldsymbol{y}}_{just} = D(\boldsymbol{z}, A(\hat{\boldsymbol{y}}_{kp}))$.

The above formulation contains two types of personalized explanations: a list of keyphrases $\hat{\boldsymbol{y}}_{kp}$ that reflects the different aspects of item i that the user u cares about (i.e., the overlap between their profiles) and a natural language explanation $\hat{\boldsymbol{y}}_{just}$ that justifies the rating, conditioned on $\hat{\boldsymbol{y}}_{kp}$. The set of keyphrases is mined from the reviews and reflects the different aspects deemed important by the users. The keyphrases enable an interaction mechanism: users can express agreement or disagreement with respect to one or multiple aspects and hence critique the recommendation.

Entangling User-Item

A key objective of T-RECS is to build a powerful latent representation. It accurately captures user and item profiles with their writing styles and entangles the rating, keyphrases, and a natural language justification. Inspired by the superiority of the Transformer for text generation tasks [Radford *et al.*, 2019], we propose a Transformer-based encoder that learns latent personalized features from users' and items' justifications. We first pass each justification J^u_j (respectively J^i_j) through the Transformer to compute the intermediate representations $\boldsymbol{h}^u_j$ (respectively $\boldsymbol{h}^i_j$). We apply a sigmoid function on the representations and average them to get $\boldsymbol{\gamma}^u$ and $\boldsymbol{\gamma}^i$:

$$\boldsymbol{\gamma}^u = \frac{1}{|J^u|}\sum\nolimits_{j \in J^u} \sigma(\boldsymbol{h}^u_j) \quad \boldsymbol{\gamma}^i = \frac{1}{|J^i|}\sum\nolimits_{j \in J^i} \sigma(\boldsymbol{h}^i_j).$$

In parallel, the encoder maps the user u (item i) to the latent factors $\boldsymbol{\beta}^u$ ($\boldsymbol{\beta}^i$) via an embedding layer. We compute the latent representation $\boldsymbol{z}$ by concatenating the latent personalized features and factors and applying a linear projection: $\boldsymbol{z} = E(u, i) = W[\boldsymbol{\gamma}^u \parallel \boldsymbol{\gamma}^i \parallel \boldsymbol{\beta}^u \parallel \boldsymbol{\beta}^i] + \boldsymbol{b}$, where $\parallel$ is the concatenation operator, and W and $\boldsymbol{b}$ the projection parameters.

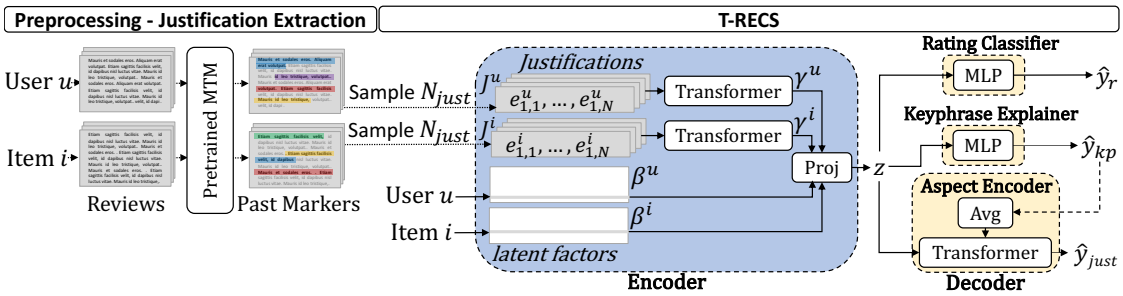


Figure 4: (Left) Preprocessing for the users and the items. For each user u and item i, we first extract *markers* from their past reviews (highlighted in color), using the pretrained multi-target masker (see Section 3), that become their respective justifications. Then, we sample N_{just} of them and build the justification references J^u and J^i, respectively. (Right) T-RECS architecture. Given a user u and an item i with their justification references J^u, J^i and latent factors β^u, β^i, T-RECS produces a joint embedding $\boldsymbol{z}$ from which it predicts a rating $\hat{\boldsymbol{y}}_r$, a keyphrase explanation $\hat{\boldsymbol{y}}_{kp}$, and a natural language justification $\hat{\boldsymbol{y}}_{just}$ conditioned on $\hat{\boldsymbol{y}}_{kp}$.

Rating Classifier & Keyphrase Explainer

Our framework classifies the interaction between the user u and item i as positive or negative. Moreover, we predict the keyphrases that describe the overlap of their profiles. Both models are a two-layer feedforward neural network with LeakyRelu activation function. Their respective losses are:

$$\mathcal{L}_r(C^r(\boldsymbol{z}), \boldsymbol{y}_r) = (\hat{\boldsymbol{y}}_r - \boldsymbol{y}_r)^2$$

$$\mathcal{L}_{kp}(C^{kp}(\boldsymbol{z}), \boldsymbol{y}_{kp}) = -\sum_{k=1}^{|K|} y_{kp}^k \log \hat{y}_{kp}^k$$

where $\mathcal{L}_r$ is the mean square error, $\mathcal{L}_{kp}$ the binary cross-entropy, and K the whole set of keyphrases.

Justification Generation

The last component consists of generating the justification. Inspired by "plan-and-write" [Yao *et al.*, 2019], we advance the personalization of the justification by incorporating the keyphrases $\hat{\boldsymbol{y}}_{kp}$. In other words, T-RECS generates a natural language justification conditioned on the 1. user, 2. item, and 3. aspects of the item that the user would consider important. We encode these via the Aspect Encoder A that takes the average of their word embeddings from the embedding layer in the Transformer. The aspect embedding is denoted by $\boldsymbol{a}_{kp}$ and added to the latent representation: $\tilde{\boldsymbol{z}} = \boldsymbol{z} + \boldsymbol{a}_{kp}$. Based on $\tilde{\boldsymbol{z}}$, the Transformer decoding block computes the output probability $\hat{y}_{just}^{t,w}$ for the word w at time-step t. We train using teacher-forcing and cross-entropy with label smoothing:

$$\mathcal{L}_{just}(D(\boldsymbol{z}, \boldsymbol{a}_{kp}), \boldsymbol{y}_{just}) = -\sum_{t=1}^{|\boldsymbol{y}_{just}|} CE(y_{just}^{t,w}, \hat{y}_{just}^{t,w})$$

We train T-RECS end-to-end and minimize jointly the loss $\boldsymbol{\mathcal{L}} = \lambda_r \mathcal{L}_r + \lambda_{kp} \mathcal{L}_{kp} + \lambda_{just} \mathcal{L}_{just}$, where λ_r, λ_{kp}, and λ_{just} control the impact of each loss. All objectives share the latent representation $\boldsymbol{z}$ and are thus mutually regularized by the function $E(u, i)$ to limit overfitting by any objective.

4.2 Unsupervised Critiquing

The purpose of critiquing is to refine the recommendation based on the user's interaction with the explanation, the keyphrases $\hat{\boldsymbol{y}}_{kp}$, represented with a binary vector. The user critiques either a keyphrase k by setting $\hat{y}_{kp}^k = 0$ (i.e., disagreement) or symmetrically adding a new one (i.e.,$\hat{y}_{kp}^k = 1$).

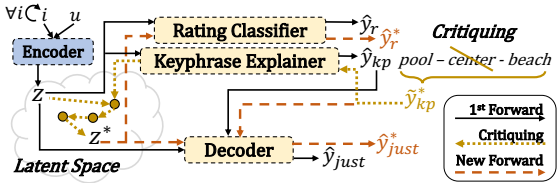


Figure 5: Workflow of considering to recommend items to a user u. We illustrate it for a given item i. **Black** denotes the forward pass to infer the rating $\hat{\boldsymbol{y}}_r$ with the explanations $\hat{\boldsymbol{y}}_{kp}$ and $\hat{\boldsymbol{y}}_{just}$. **Yellow** indicates the critiquing: the user critiques the binary-vector keyphrase explanation $\hat{\boldsymbol{y}}_{kp}$ (e.g., *center*) to $\tilde{\boldsymbol{y}}_{kp}^*$, which modulates the latent space into $\boldsymbol{z}^*$ for each item. **Orange** shows the new forward pass for the subsequent recommendation $\hat{\boldsymbol{y}}_r^*$ and explanations $\hat{\boldsymbol{y}}_{kp}^*$, $\hat{\boldsymbol{y}}_{just}^*$.

We denote the critiqued keyphrase explanation as $\tilde{\boldsymbol{y}}_{kp}^*$. The overall critiquing process is depicted in Fig. 5. Inspired by the recent success in editing the latent space on the unsupervised text attribute transfer task [Wang *et al.*, 2019], we employ the trained Keyphrase Explainer C^{kp} and the critiqued explanation $\tilde{\boldsymbol{y}}_{kp}^*$ to provide the gradient from which we update the latent representation $\boldsymbol{z}$ (depicted in **yellow**). More formally, given a latent representation $\boldsymbol{z}$ and a binary critique vector $\tilde{\boldsymbol{y}}_{kp}^*$, we want to find a new latent representation $\boldsymbol{z}^*$ that will produce a new keyphrase explanation close to the critique, such that $|C^{kp}(\boldsymbol{z}^*) - \tilde{\boldsymbol{y}}_{kp}^*| \leq T$, where T is a threshold. In order to achieve this goal, we iteratively compute the gradient with respect to $\boldsymbol{z}$ instead of the model parameters C_θ^{kp}. We then modify $\boldsymbol{z}$ in the direction of the gradient until we get a new latent representation $\boldsymbol{z}^*$ that C^{kp} considers close enough to $\tilde{\boldsymbol{y}}_{kp}^*$ (shown in **orange**). We emphasize that we use the gradient to modulate $\boldsymbol{z}$ rather than the parameters C^{kp}.

Let denote the gradient as $\boldsymbol{g}_t$ and a decay coefficient as ζ. For each iteration t and $\boldsymbol{z}_0^* = \boldsymbol{z}$, the modified latent representation $\boldsymbol{z}_t^*$ at the t^{th} iteration can be formulated as follows:

$$\boldsymbol{g}_t = \nabla_{\boldsymbol{z}_t^*} \mathcal{L}_{kp}(C^{kp}(\boldsymbol{z}_t^*), \tilde{\boldsymbol{y}}_{kp}^*); \;\; \boldsymbol{z}_t^* = \boldsymbol{z}_{t-1}^* - \zeta^{t-1} \boldsymbol{g}_t / ||\boldsymbol{g}_t||_2$$

Because this optimization is nonconvex, there is no guarantee that the difference between the critique vector and the inferred explanation will differ by only T. In our experiments in Section 5.4, we found that a limit of 50 iterations works well, and that the newly induced explanations remain consistent.

5 Experiments

5.1 Experimental Settings

Datasets. We evaluate the quantitative performance of T-RECS using two real-world, publicly available datasets: BeerAdvocate [McAuley and Leskovec, 2013] and HotelRec [Antognini and Faltings, 2020]. They contain 1.5 and 50 million reviews from BeerAdvocate and TripAdvisor. In addition to the overall rating, users also provided five-star aspect ratings. We binarize the ratings with a threshold t: $t > 4$ for hotel reviews and $t > 3.5$ for beer reviews. We further filter out all users with fewer than 20 interactions and sort them chronologically. We keep the first 80% of interactions per user as the training data, leaving the remaining 20% for validation and testing. We sample two justifications per review. We need to select keyphrases for explanations and critiquing. Hence, we follow the processing in [Wu *et al.*, 2019] to extract 200 keyphrases (distributed uniformly over the aspect categories) from the *markers* on each dataset.

Implementation Details. To extract *markers*, we trained MTM with the hyperparameters reported by the authors. We build the justification history J^u,J^i, with $N_{just} = 32$. We set the embedding and attention dimension to 256 and to 1024 for the feed-forward network. The encoder and decoder consist of two layers of Transformer with 4 attention heads. We use a batch size of 128, dropout of 0.1, and Adam with learning rate 0.001. For critiquing, we choose a threshold and decay coefficient $T = 0.015, \zeta = 0.9$ and $T = 0.01, \zeta = 0.975$ for hotel and beer reviews. We tune all models on the dev set. For reproducibility purposes, we provide details in Appendix.[1]

5.2 RQ 1: Are *Markers* Appropriate Justifications?

We derive baselines from [Ni *et al.*, 2019]: we split a review into elementary discourse units (EDUs) and apply their classifier to get justifications; it is trained on a manually annotated dataset and generalizes well to other domains. We employ two variants: EDU One and EDU All. The latter includes all justifications, whereas the former includes only one.

We perform a human evaluation using Amazon's Mechanical Turk (see Appendix for more details) to judge the quality of the justifications extracted from the Markers, EDU One, and EDU All on both datasets. We employ three setups: an evaluator is presented with 1. the three types of justifications; 2. only those from Markers and EDU All; and 3. EDU One instead of EDU All. We sampled 300 reviews (100 per setup) with generated justifications presented in random order. The annotators judged the justifications by choosing the most convincing in the pairwise setups and otherwise using best-worst scaling. We report the win rates for the pairwise comparisons and a normalized score ranging from -1 to +1.

Table 2 shows that justifications extracted from Markers are preferred, on both datasets, more than 80% of the time. Moreover, when compared to EDU All and EDU One, Markers achieve a score of 0.74, three times higher than EDU All. Therefore, justifications extracted from the Markers are significantly better than EDUs, and a single justification cannot explain a review. Fig. 3 shows a sample for comparison.

[1]Appendices are available at http://arxiv.org/pdf/2005.11067.pdf

Dataset	#Users	#Items	#Inter.	Dens.	KP Cov.	Avg. #KP per Just.	Avg. #KP per Rev.	Avg. #KP per User
Hotel	72,603	38 896	2.2M	0.08%	97.66%	2.15	3.79	115
Beer	7,304	8,702	1.2M	2.02%	96.87%	3.72	6.97	1,210

Table 1: Descriptive statistics of the datasets.

	Hotel			*Beer*		
Winner Loser	**Win Rate**			**Win Rate**		
Markers EDU All	81%**			77%**		
Markers EDU One	93%**			90%**		
Model	**Score**	**#B**	**#W**	**Score**	**#B**	**#W**
EDU One	-0.95**	1	96	-0.93**	2	95
EDU All	0.21**	24	3	0.20**	23	3
Markers	**0.74**	75	1	**0.73**	75	2

Table 2: Human evaluation of explanations in terms of the win rate and the **b**est-**w**orst scaling. A score significantly different than Markers (post hoc Tukey HSD test) is denoted by ** for $p < 0.001$.

5.3 RQ 2: Does T-RECS Generate High-Quality, Relevant, and Personalized Explanations?

Natural Language Explanations. We consider five baselines: ExpansionNet [Ni and McAuley, 2018] is a seq-to-seq model with a user, item, aspect, and fusion attention mechanism that generates personalized reviews. DualPC [Sun *et al.*, 2020] and CAML [Chen *et al.*, 2019] generate an explanation based on a rating and the user-item pair. Ref2Seq improves upon ExpansionNet by learning only from historical justifications of a user and an item. AP-Ref2Seq [Ni *et al.*, 2019] extends Ref2Seq with aspect planning [Yao *et al.*, 2019], in which aspects are given during the generation. All models use beam search during testing and the same keyphrases as aspects. We employ BLEU, ROUGE-L, BertScore [Zhang *et al.*, 2020], the perplexity for the fluency, and R_{KW} for the explanation consistency as in [Chen *et al.*, 2020]: the ratio of the target keyphrases present in the generated justifications.

The main results are presented in Table 3 (more in Appendix). T-RECS achieves the highest scores on both datasets. We note that 1. seq-to-seq models better capture user and item information to produce more relevant justifications, and 2. using a keyphrase plan doubles the performance on average and improving explanation consistency.

We run a human evaluation, with the best models according to R_{KW}, using best-worst scaling on the dimensions: overall, fluency, informativeness, and relevance. We sample 300 explanations and showed them in random order. Table 4 shows that our explanations are largely preferred on all criteria.

Keyphrase Explanations. We compare T-RECS with the popularity baseline and the models proposed in [Wu *et al.*, 2019], which are extended versions of the NCF model [He *et al.*, 2017]. E-NCF and CE-NCF augment the NCF method with an explanation and a critiquing neural component. Also, the authors provide variational variants: VNCF, E-VNCF, and CE-VNCF. Here, we omit NCF and VNCF because they are trained only to predict ratings. We report the following metrics: NDCG, MAP, Precision, and Recall at 10.

	Model	BLEU	R-L	$\text{BERT}_{\text{Score}}$	PPL↓	R_{KW}
Hotel	ExpansionNet	0.53	6.91	74.81	28.87	60.09
	DualPC	1.53	16.73	86.76	28.99	13.12
	CAML	1.13	16.67	87.77	29.10	9.38
	Ref2Seq	1.77	16.45	86.74	29.07	13.19
	AP-Ref2Seq	7.28	33.71	88.31	21.31	90.20
	T-RECS	**7.47**	**34.10**	**90.23**	**17.80**	**93.57**
Beer	ExpansionNet	1.22	9.68	72.32	22.28	82.49
	DualPC	2.08	14.68	85.49	21.15	10.60
	CAML	2.43	14.99	85.96	21.29	10.18
	Ref2Seq	3.51	15.96	85.27	22.34	12.10
	AP-Ref2Seq	15.89	46.50	91.35	12.07	91.52
	T-RECS	**16.54**	**47.20**	**91.50**	**10.24**	**94.96**

Table 3: Generated justifications on automatic evaluation.

	Hotel				*Beer*			
Model	**O**	**F**	**I**	**R**	**O**	**F**	**I**	**R**
ExpansionNet	-0.58	-0.67	-0.52	-0.56	-0.03	-0.31	0.10	-0.01
Ref2Seq	-0.27	-0.19	-0.30	-0.26	-0.69	-0.34	-0.71	-0.69
AP-Ref2Seq	0.30	0.32	0.29	0.29	0.22	0.25†	0.21†	0.25
T-RECS	**0.55**	**0.54**	**0.53**	**0.53**	**0.49**	**0.39**	**0.39**	**0.45**

Table 4: Human evaluation of justifications in terms of best-worst scaling for **O**verall, **F**luency, **I**nformativenss, and **R**elevance. Most scores are significantly different than T-RECS (post hoc Tukey HSD test) with $p < 0.002$. † denotes a nonsignificant score.

Table 5 shows that T-RECS outperforms the CE-(V)NCF models by 60%, Pop by 20%, and E-(V)NCF models by 10% to 30% on all datasets. Pop performs better than CE-(V)NCF, showing that many keywords are recurrent in reviews. Thus, predicting keyphrases from the user-item latent space is a natural way to entangle them with (and enable critiquing).

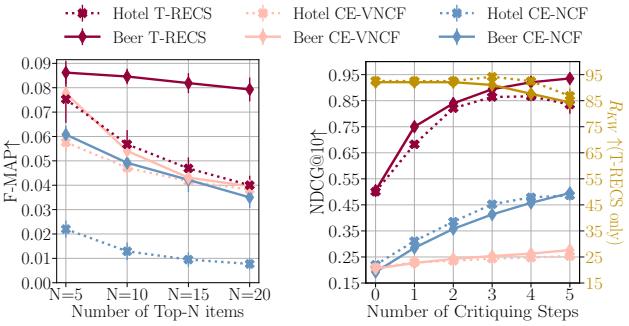
5.4 RQ 3: Can T-RECS Enable Critiquing?

Single-Step Critiquing. For a given user, T-RECS recommends an item and generates personalized explanations, where the user can interact by critiquing one or multiple keyphrases. However, no explicit ground truth exists to evaluate the critiquing. We use F-MAP [Wu *et al.*, 2019] to measure the effect of a critique. Given a user, a set of recommended items $\mathbb{S}$, and a critique k, let $\mathbb{S}_k$ be the set of items containing k in the explanation. The F-MAP measures the ranking difference of the affected items $\mathbb{S}_k$ before and after critiquing k, using the Mean Average Precision at N. A positive F-MAP indicates that the rank of items in $\mathbb{S}_k$ fell after k is critiqued. We compare T-RECS with CE-(V)NCF and average the F-MAP over $5,000$ user-keyphrase pairs.
Fig. 6a presents the F-MAP performance on both datasets. All models show an anticipated positive F-MAP. The performance of T-RECS improves considerably on the beer dataset and is significantly higher for $N \leq 10$ on the hotel dataset. The gap in performance may be caused by the extra loss of the autoencoder, which brings noise during training. T-RECS only iteratively edits the latent representation at test time.

Multi-Step Critiquing. Evaluating multi-step critiquing via ranking is difficult because many items can have the keyphrases of the desired item. Instead, we evaluate whether a system obtains a complete model of the user's preferences following [Pu *et al.*, 2006]. A user expresses his keyphrase preferences iteratively according to a randomly selected liked item. After each step, we evaluate the keyphrase explanations. For T-RECS, we also report the explanation consistency R_{KW}. We run up to five-steps critiques over $1,000$ random selected users and up to $5,000$ random keyphrases for each dataset. Fig. 6b shows that T-RECS builds through the critiques more accurate user profiles and consistent explanations. CE-NCF's top performance is significantly lower than T-RECS, and CE-VNCF plateaus, surely because of the KL divergence regularization, which limits the amount of information stored in the latent space. The explanation quality in T-RECS depends on the accuracy of the user's profile and may become saturated once we find it after four steps.[2]

	Hotel				*Beer*			
Model	**NDCG**	**MAP**	**P**	**R**	**NDCG**	**MAP**	**P**	**R**
Pop	0.333	0.208	0.143	0.396	0.250	0.229	0.176	0.253
E-NCF	0.341	0.215	0.137	0.380	0.249	0.220	0.179	0.262
CE-NCF	0.229	0.143	0.092	0.255	0.192	0.172	0.136	0.197
E-VNCF	0.344	0.216	0.139	0.386	0.236	0.210	0.170	0.248
CE-VNCF	0.229	0.134	0.107	0.297	0.203	0.178	0.148	0.215
T-RECS	**0.376**	**0.236**	**0.158**	**0.436**	**0.316**	**0.280**	**0.228**	**0.332**

Table 5: Keyphrase explanation quality at $N = 10$.



(a) Falling MAP for different top-N. Error bars show the standard deviation.

(b) Keyphrase prediction over multi-step critiquing with 95% confidence interval. We also report the explanation consistency R_{KW} for T-RECS.

Figure 6: Single- (top) and multi-step (bottom) critiquing.

6 Conclusion

Recommendations can carry much more impact if they are supported by explanations. We presented T-RECS, a multi-task learning Transformer-based recommender, that produces explanations considered significantly superior when evaluated by humans. The second contribution of T-RECS is the user's ability to react to a recommendation by *critiquing* the explanation. We designed an unsupervised method for multi-step critiquing with explanations. Experiments show that T-RECS obtains stable and significant improvement in adapting to the preferences expressed in multi-step critiquing.

[2]We could not compare T-RECS with [Chen *et al.*, 2020] because the authors did not make the code available due to copyright issues.

References

[Antognini and Faltings, 2020] Diego Antognini and Boi Faltings. Hotelrec: a novel very large-scale hotel recommendation dataset. In *the Language Resources and Evaluation Conference*, 2020.

[Antognini *et al.*, 2021] Diego Antognini, Claudiu Musat, and Boi Faltings. Multi-dimensional explanation of target variables from documents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35, 2021.

[Burke *et al.*, 1996] Robin D. Burke, Kristian J. Hammond, and Benjamin C. Young. Knowledge-based navigation of complex information spaces. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, page 462–468, 1996.

[Catherine and Cohen, 2017] Rose Catherine and William Cohen. Transnets: Learning to transform for recommendation. In *Proceedings of the ACM conference on recommender systems*, 2017.

[Chen and Pu, 2012] Li Chen and Pearl Pu. Critiquing-based recommenders: survey and emerging trends. *User Modeling and User-Adapted Interaction*, 22(1-2), 2012.

[Chen *et al.*, 2019] Zhongxia Chen, Xiting Wang, Xing Xie, Tong Wu, Guoqing Bu, Yining Wang, and Enhong Chen. Co-attentive multi-task learning for explainable recommendation. In *IJCAI*, pages 2137–2143, 2019.

[Chen *et al.*, 2020] Zhongxia Chen, Xiting Wang, Xing Xie, Mehul Parsana, Akshay Soni, Xiang Ao, and Enhong Chen. Towards explainable conversational recommendation. IJCAI, 2020.

[Dong *et al.*, 2017] Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. Learning to generate product reviews from attributes. In *the Conference of the European Association for Computational Linguistics*, pages 623–632, 2017.

[He *et al.*, 2017] 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.

[Kunkel *et al.*, 2018] Johannes Kunkel, Tim Donkers, Catalin-Mihai Barbu, and Jürgen Ziegler. Trust-related effects of expertise and similarity cues in human-generated recommendations. In *2nd Workshop on Theory-Informed User Modeling*, 2018.

[Kunkel *et al.*, 2019] Johannes Kunkel, Tim Donkers, Lisa Michael, Catalin-Mihai Barbu, and Jürgen Ziegler. Let me explain: Impact of personal and impersonal explanations on trust in recommender systems. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2019.

[Li *et al.*, 2017] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. Neural rating regression with abstractive tips generation for recommendation. In *the ACM SIGIR conference on Research and Development in Information Retrieval*, 2017.

[Li *et al.*, 2019] Piji Li, Zihao Wang, Lidong Bing, and Wai Lam. Persona-aware tips generation. In *The World Wide Web Conference*, 2019.

[Luo *et al.*, 2020] Kai Luo, Scott Sanner, Ga Wu, Hanze Li, and Hojin Yang. Latent linear critiquing for conversational recommender systems. In *Proceedings of the 29th International Conference on the World Wide Web*, 2020.

[Mann and Thompson, 1988] William C Mann and Sandra A Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281, 1988.

[McAuley and Leskovec, 2013] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *the ACM conference on Recommender systems*, pages 165–172. ACM, 2013.

[Musat and Faltings, 2015] Claudiu Cristian Musat and Boi Faltings. Personalizing product rankings using collaborative filtering on opinion-derived topic profiles. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[Ni and McAuley, 2018] Jianmo Ni and Julian McAuley. Personalized review generation by expanding phrases and attending on aspect-aware representations. In *Proceedings of the Association for Computational Linguistics*, pages 706–711, 2018.

[Ni *et al.*, 2019] Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, 2019.

[Pu *et al.*, 2006] Pearl Pu, Paolo Viappiani, and Boi Faltings. Increasing user decision accuracy using suggestions. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 121–130, 2006.

[Radford *et al.*, 2019] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

[Reilly *et al.*, 2005] James Reilly, Kevin McCarthy, Lorraine McGinty, and Barry Smyth. Explaining compound critiques. *Artificial Intelligence Review*, 2005.

[Sun *et al.*, 2020] 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 World Wide Web*, 2020.

[Tintarev and Masthoff, 2015] Nava Tintarev and Judith Masthoff. Explaining recommendations: Design and evaluation. In *Recommender systems handbook*, pages 353–382. Springer, 2015.

[Wang *et al.*, 2019] Ke Wang, Hang Hua, and Xiaojun Wan. Controllable unsupervised text attribute transfer via editing entangled latent representation. In *Annual Conference on Neural Information Processing Systems*, pages 11034–11044, 2019.

[Williams and Tou, 1982] Michael D Williams and Frederich N Tou. Rabbit: an interface for database access. In *Proceedings of the ACM Conference*, pages 83–87, 1982.

[Wu *et al.*, 2019] Ga Wu, Kai Luo, Scott Sanner, and Harold Soh. Deep language-based critiquing for recommender systems. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 137–145, 2019.

[Yao *et al.*, 2019] Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7378–7385, 2019.

[Zhang and Curley, 2018] Jingjing Zhang and Shawn P Curley. Exploring explanation effects on consumers' trust in online recommender agents. *International Journal of Human–Computer Interaction*, 34(5):421–432, 2018.

[Zhang *et al.*, 2014] 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 *the 37th ACM SIGIR conference on Research & development in information retrieval*, pages 83–92, 2014.

[Zhang *et al.*, 2020] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *ICLR 2020*, 2020.