Deep Language-based Critiquing for Recommender Systems

Ga Wu*†
University of Toronto
wuga@mie.utoronto.ca

Kai Luo†
University of Toronto
kluo@mie.utoronto.ca

Scott Sanner*
University of Toronto
ssanner@mie.utoronto.ca

Harold Soh
National University of Singapore
harold@comp.nus.edu.sg

ABSTRACT

Critiquing is a method for conversational recommendation that adapts recommendations in response to user preference feedback regarding item attributes. Historical critiquing methods were largely based on constraint- and utility-based methods for modifying recommendations w.r.t. these critiqued attributes. In this paper, we revisit the critiquing approach from the lens of deep learning based recommendation methods and language-based interaction. Concretely, we propose an end-to-end deep learning framework with two variants that extend the Neural Collaborative Filtering architecture with explanation and critiquing components. These architectures not only predict personalized keyphrases for a user and item but also embed language-based feedback in the latent space that in turn modulates subsequent critiqued recommendations. We evaluate the proposed framework on two recommendation datasets containing user reviews. Empirical results show that our modified NCF approach not only provides a strong baseline recommender and high-quality personalized item keyphrase suggestions, but that it also properly suppresses items predicted to have a critiqued keyphrase. In summary, this paper provides a first step to unify deep recommendation and language-based feedback in what we hope to be a rich space for future research in deep critiquing for conversational recommendation.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → *Neural networks.*

KEYWORDS

Deep Learning; Conversational Recommendation; Critiquing

ACM Reference Format:
Ga Wu, Kai Luo, Scott Sanner, and Harold Soh. 2019. Deep Language-based Critiquing for Recommender Systems. In *Thirteenth ACM Conference on Recommender Systems (RecSys '19), September 16–20, 2019, Copenhagen, Denmark.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3298689.3347009

*Affiliate to Vector Institute of Artificial Intelligence, Toronto
†Both authors contributed equally to this work

RecSys '19, September 16–20, 2019, Copenhagen, Denmark

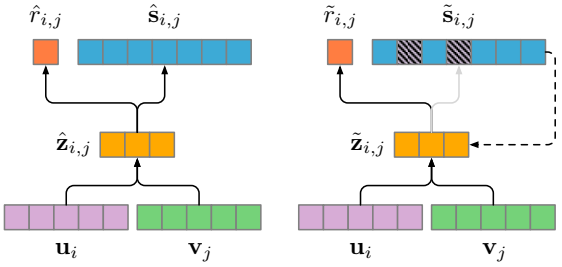
ACM ISBN 978-1-4503-6243-6/19/09…$15.00
https://doi.org/10.1145/3298689.3347009

1 INTRODUCTION

Critiquing is a method for conversational (a.k.a. sequential interactive) recommendation that adapts recommendations in response to user preference feedback regarding item attributes. For example, in unit critiquing [2], a user might critique a digital camera recommendation by requesting an item with higher resolution and in compound critiquing [19, 21], a user might further explore items that have longer battery life *and* lower price than an initial recommendation. Further extensions such as incremental critiquing consider the cumulative effect of iterated critiquing interactions [20] while experience-based methods attempt to collaboratively leverage critiquing interactions from multiple users [17].

While previous work on critiquing has collectively defined an important subfield of conversational recommender systems, most of these methods assume a fixed set of known attributes along with explicit constraint- and utility-based methods for modifying recommendations w.r.t. critiqued attributes. Though some work has focused on explanations in critiquing [21] and other work [8, 24] has respectively explored speech- and dialog-based interfaces for critiquing-style frameworks, these architectures have nonetheless assumed that item attributes are explicitly known *a priori.*

Furthermore, it is important to note that (conversational) recommendation research has progressed substantially from the underlying techniques that drove historical critiquing-based methods, many of which are over a decade old [7]. First, from a basic recommendation perspective, deep-learning based recommendation methods currently produce state-of-the-art results [10, 14, 23, 27]. Second, recent efforts have focused on conversational recommendation methods that do not assume fixed *a priori* item attributes, but rather actively apply explore-exploit strategies on top of latent factor models [3, 30]. Third, the use of language-based explanations has also advanced significantly [28]; to name just a few of these works: McAuley et al. [15] introduces topic extraction methods to explain and highlight key aspects of recommended items, Zhang et al. [29] improves explanation understandability by filling key features of recommended items into template sentences, and Costa et al. [4] and Li et al. [13] both directly generate text explanations for recommendations using Recurrent Neural Networks. Yet, given all of these individual advances in deep recommendation, latent factor models, and explanation, we are not aware of work that has combined them in a deep conversational critiquing framework.



(a) Explanation **(b) Critiquing**

Figure 1: Proposed CE-(V)NCF architecture. (a) Given user $\mathbf{u}_i$ and item $\mathbf{v}_j$ embeddings as input, the network produces a joint embedding $\hat{\mathbf{z}}_{i,j}$ and an initial rating $\hat{r}_{i,j}$ and explanation $\hat{\mathbf{s}}_{i,j}$ via forward propagation. (b) Shaded squares indicate critiqued keyphrase explanations that modulate the latent space into $\tilde{\mathbf{z}}_{i,j}$ for subsequent recommendations.

In this work, we aim to revisit the critiquing framework from the lens of deep-learning based recommendation methods as well as language-based interaction that can work with fixed a set of *inferred* keyphrase attributes. These framework modifications necessitate two major changes in how we approach critiquing: (1) In deep learning based systems, user preferences and feedback must both be represented and manipulated in the *same latent embedded space* to provide updated recommendations after critiquing; (2) Language-based interaction allows for a richer space of interaction than a set of fixed item attributes, but also introduces issues with subjective (personalized) judgments of language-based labels, data sparsity, synonymity, and inherent label uncertainty.

To address issues (1) and (2), we begin by restricting the language interaction to a large set of descriptive keyphrases mined from user reviews. We then propose an end-to-end deep learning framework with two variants – one deterministic and one probabilistic – that extend Neural Collaborative Filtering (NCF) [10] with explanation and critiquing components. These architectures not only infer *personalized* keyphrase explanations for a user and item but also embed language-based feedback in the same latent space as user and item embeddings in order to modulate subsequent critiqued recommendations.

We evaluate this deep language-based critiquing framework on two recommendation datasets containing user reviews. We observe that our modified NCF approach not only provides a strong baseline recommender and high-quality personalized item keyphrase suggestions, but that it also properly suppresses items predicted to have a critiqued keyphrase. We further demonstrate how the variational probabilistic approach we propose yields the most compatible co-embeddings of user and item preferences with language-based critiques due to its KL-divergence regularization of the latent space. In summary, this paper provides a modern update of the critiquing framework to combine deep recommendation with language-based feedback in what we hope to be a rich space for future research.

2 DEEP EXPLANATION AND CRITIQUING

In this paper, we propose an end-to-end deep learning architecture that produces human understandable recommendation explanations and allows users to critique these generated explanations to refine the recommendations (Figure 1). We present two variants of the proposed model: one deterministic and one probabilistic. Our discussion begins with the deterministic variant, which provides the basic intuition underlying our approach. We then generalize the deterministic model into a variational probabilistic version that offers better performance for an additional computational cost.

Table 1: Example keyphrases extracted from the review datasets. We categorize the reasons for better understanding of the extraction only.

Dataset	Reason Type	Keyphrases
Beer	Head	white, tan, offwhite, brown
	Malt	roasted, caramel, pale, wheat, rye
	Color	golden, copper, orange, black, yellow
	Taste	citrus, fruit, chocolate, cherry, plum
CDs&Vinyl	Genre	rock, pop, jazz, rap, hip hop, R&B
	Instrument	orchestra, drum
	Style	concert, opera
	Religious	chorus, christian, gospel

2.1 Deterministic Model: CE-NCF

We present the deterministic model in three parts: explanation generation, explanation critiquing, and the overall training objective.

2.1.1 Explanation Generation. The key hypothesis of our explainable model is that the observed user i and item j ratings $r_{i,j} \in \{0 \text{ (dislike)}, 1 \text{ (like)}\}$ and binary explanation vector $\mathbf{s}_{i,j}$ are generated from the same underlying latent representation $\mathbf{z}$ encoded jointly from the latent user $\mathbf{u}_i$ and item $\mathbf{v}_j$ representations.

We formulate the hypothesis into a deep-learning framework as shown in Figure 1(a), in which the model is first trained to encode a user embedding $\mathbf{u}_i$ and item embedding $\mathbf{v}_j$ pair into an initial latent representation $\hat{\mathbf{z}}_{i,j}$ via an encoding function f_e. Then, the prediction function f_r and f_s generate the probability of a "like" interaction $\hat{r}_{i,j} \in [0, 1]$ and explanation $\hat{\mathbf{s}}_{i,j}$ for this user-item pair, respectively. To summarize, our model consists of three functions:

$$\hat{\mathbf{z}}_{i,j} = f_e(\mathbf{u}_i, \mathbf{v}_j), \quad \hat{\mathbf{s}}_{i,j} = f_s(\hat{\mathbf{z}}_{i,j}), \quad \text{and} \quad \hat{r}_{i,j} = f_r(\hat{\mathbf{z}}_{i,j}). \quad (1)$$

The above formulation is general in terms of explanation types; the explanation could be a list of similar items or stated reasons in natural language. In this paper, we use a list of keyphrases extracted from reviews since they are informative and interpretable, i.e., the keyphrases we will mine from user reviews are intended to reflect precise reasons for liking or disliking an item. As such, keyphrases also support a simple interaction mechanism that enables users to express disagreement (or agreement) with a personalized keyphrase explanation for the recommendation and hence, critique the recommendation. Table 1 shows some examples of the extracted keyphrases from the BeerAdvocate and Amazon CDs&Vinyl review datasets we experiment with in this paper.

2.1.2 Explanation Critiquing. The purpose of explanation critiquing is to refine the recommendation based on a user's interaction with the explanations. Intuitively, critiquing enables users to correct the "static" tastes learned during training so that the recommendation system can better match current user preferences.

In our model (Figure 1(b)), the critiquing process augments the latent representation, which in turn modifies the ratings to better

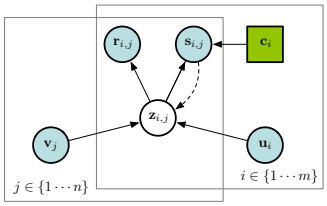


Figure 2: Probabilistic Graphical Model view of the proposed CE-(V)NCF model. Action node $\mathbf{c}_i$ represents a critiquing action of user i that modifies the predicted explanation $\mathbf{s}_{i,j}$ into critiqued explanation $\tilde{\mathbf{s}}_{i,j}$. The dashed arrow denotes posterior inference after critiquing.

suit the user's current preferences. Technically, this can be achieved via an inverse function,

$$\tilde{\mathbf{z}}_{i,j} = f_s^{-1}(\tilde{\mathbf{s}}_{i,j}) \tag{2}$$

where $\tilde{\mathbf{s}}_{i,j}$ represents the critiqued explanation and f_s^{-1} is the inverse function of f_s. Unfortunately, the inverse function f_s^{-1} may not exist in general. Inspired by the autoencoder [1], we propose a work-around by learning an approximation $\tilde{f}_s^{-1}$ through minimizing the reconstruction loss of the latent representation.

As previously noted, the predicted explanations can be critiqued by "disagreeing" with a subset of keyphrases as demonstrated in Figure 1. Precisely, the critiquing steps are summarized as follows:

(1) The prediction function $f_s(\mathbf{z})$ maps the latent representation into rating and explanation predictions for each recommended item to a particular user.
(2) The user takes a critiquing action by indicating which explanations they disagree with, effectively "zeroing out" these keyphrases in $\hat{\mathbf{s}}_{i,j}$.
(3) An inverse prediction function $f_s^{-1}(\mathbf{z})$ projects the critiqued explanation back to the latent representation.
(4) Finally, the model updates the rating and explanation for each user-item pair as follows:

$$\tilde{\mathbf{s}}_{i,j} = f_s(\tilde{\mathbf{z}}_{i,j}), \quad \text{and} \quad \tilde{r}_{i,j} = f_r(\tilde{\mathbf{z}}_{i,j})\,. \tag{3}$$

To re-rank items based on the new rating predictions, critiquing is applied to explanations of all items recommended to the user.

In addition, to flexibly control the degree of critiquing, we smooth critiqued updates to the latent space as a linear combination of the critiqued latent representation and the initial latent representation:

$$\tilde{\mathbf{z}}_{i,j} = \rho\hat{\mathbf{z}}_{i,j} + (1-\rho)\tilde{\mathbf{z}}_{i,j}\,, \tag{4}$$

where $\rho \in [0,1]$ represents a hyperparameter that balances the impact of the critique on the initial latent representation.

2.1.3 Training Objective. We now train the previously discussed framework end-to-end by jointly minimizing the objective

$$\begin{aligned}\min \mathcal{L} = \min & \sum_{i,j} \mathcal{L}_0(r_{i,j}, f_r \circ f_e(\mathbf{u}_i, \mathbf{v}_j)) \\ & + \lambda_1 \sum_{i,j} \mathcal{L}_1(\mathbf{s}_{i,j}, f_s \circ f_e(\mathbf{u}_i, \mathbf{v}_j)) \\ & + \lambda_2 \sum_{i,j} \mathcal{L}_2(f_e(\mathbf{u}_i, \mathbf{v}_j), \tilde{f}_s^{-1} \circ f_s \circ f_e(\mathbf{u}_i, \mathbf{v}_j)) \\ & + \lambda_3 ||\theta||_2^2\,,\end{aligned} \tag{5}$$

where $\mathcal{L}_0$ represents the rating prediction loss, $\mathcal{L}_1$ represents the explanation prediction loss, $\mathcal{L}_2$ represents the latent representation loss, and θ represents the trainable parameters in the encoding and prediction functions. λ_1, λ_2, and λ_3 are hyperparameters.

Intuitively, $\mathcal{L}_0$ and $\mathcal{L}_1$ are supervised objectives where the latent features $\mathbf{u}_i$, $\mathbf{v}_j$ are pretrained (cf. next section) and the labels $\mathbf{s}_{i,j}$ and $r_{i,j}$ are given. $\mathcal{L}_2$ is an autoencoding objective which encourages the model to learn latent user-item representations that are both explainable and recoverable from the explanation inverse, i.e., $\tilde{f}_s^{-1} \circ f_s$. All three objectives share the latent encoding function f_e, which serves as a strong mutual regularizer to limit overfitting of parameters in f_e by any one objective. Moreover, the loss functions $\mathcal{L}_1$ and $\mathcal{L}_2$ share the explanation generation function f_s that form another mutual regularizer over the parameters of f_s.

We refer to this deterministic model as Critiquable and Explainable Neural Collaborative Filtering (CE-NCF). While the CE-NCF model supports arbitrary loss functions $\mathcal{L}_0$, $\mathcal{L}_1$ and $\mathcal{L}_2$, we simply use Mean Squared Error (MSE) for all three objectives since it keeps the range of $\mathcal{L}_2$ compatible with $\mathcal{L}_0$ and $\mathcal{L}_1$, and proved to be more stable in training than mixing MSE with cross-entropy losses.

2.1.4 User and Item Embeddings. In previous sections, we assumed that the user and item representations are accessible as pretrained embeddings. While there are many ways to pretrain the embeddings (see [18, 26]), in this paper, we apply randomized singular value decomposition (SVD) [6] on the user-item interaction matrix

$$U = P\Sigma^{\frac{1}{2}} \quad V = Q\Sigma^{\frac{1}{2}} \quad P\Sigma Q^T = SVD(R) \tag{6}$$

to initialize the embeddings. However, since the network jointly predicts both the user-item interaction probability and the corresponding explanations, fixed SVD embeddings learned purely from interaction matrix are unlikely to yield accurate joint predictions. As such, the embeddings are fine-tuned during end-to-end training.

2.2 Variational Probabilistic Model: CE-VNCF

In this section, we generalize the proposed deterministic CE-NCF model to a probabilistic generative model (Figure 2). Concretely, we maximize the amortized variational lower-bound of the following log marginal likelihood:[1]

$$\begin{aligned}&\log \prod_{i,j} p(r_{i,j}, \mathbf{s}_{i,j} | \mathbf{u}_i, \mathbf{v}_j, \tilde{\mathbf{s}}_{i,j}) \\ &\geq \sum_{i,j} \underbrace{E_{q(\mathbf{z_{i,j}}|\mathbf{u}_i,\mathbf{v}_j)}[\log p(r_{i,j}|\mathbf{z_{i,j}})]}_{\mathcal{L}_0} + \sum_{i,j} \underbrace{E_{q(\mathbf{z_{i,j}}|\mathbf{u}_i,\mathbf{v}_j)}[\log p(\mathbf{s}_{i,j}|\mathbf{z_{i,j}})]}_{\mathcal{L}_1} \\ &\quad + \sum_{i,j} \underbrace{E_{q(\mathbf{z_{i,j}}|\mathbf{u}_i,\mathbf{v}_j)}[\log p(\mathbf{z_{i,j}}|\tilde{\mathbf{s}}_{i,j})]}_{\mathcal{L}_2} - \sum_{i,j} KL[q(\mathbf{z_{i,j}}|\mathbf{u}_i,\mathbf{v}_j)||p(\mathbf{z_{i,j}})] \\ &\quad + \lambda||\theta||_2^2,\end{aligned} \tag{7}$$

where latent distributions $p(\mathbf{z_{i,j}})$, $q(\mathbf{z_{i,j}}|\mathbf{u}_i, \mathbf{v}_j)$, and $p(\mathbf{z_{i,j}}|\tilde{\mathbf{s}}_{i,j})$ are assumed to be Gaussian. On the feedforward pass, we sample $\mathbf{z_{i,j}}$ conditioned on $\mathbf{u}_i$ and $\mathbf{v}_j$; we then set $\tilde{\mathbf{s}}_{i,j} = f_s(\mathbf{z}_{i,j})$ required to determine the loss in $\mathcal{L}_2$ and complete a gradient step. The effect of setting the explanation equal to the critique encourages invertibility of the explanation and critiquing process – *if* the

[1] A full derivation of (7) and training details are given in the online Appendix.

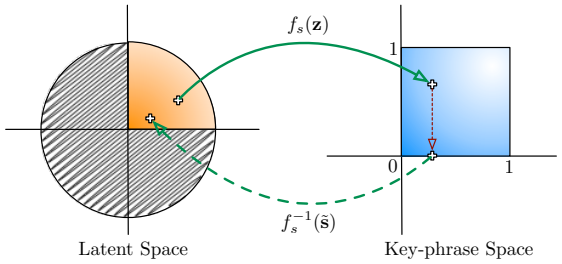


Figure 3: Conceptualization of the mapping between a 2D latent representation and a 2D keyphrase representation. The green solid line shows the latent to keyphrase space mapping and the dashed line shows the inverse after a critique zeroes out a keyphrase (red dashed projection to axis).

critique was the same as the explanation then we should recover the same embedding that produced the explanation to begin with. During variational training, all parameters of p and q are learned and the SVD-initialized embeddings $\mathbf{u}_i$ and $\mathbf{v}_j$ are fine-tuned.

We call this probabilistic extension the Critiquable and Explainable Variational Neural Collaborative Filtering (CE-VNCF). The network architecture of the probabilistic extension is identical to the deterministic model, and the three expectation terms in Equation 7 correspond to the three components of the deterministic objective in Equation 5, where the conditional probability $q(\mathbf{z_{i,j}}|\mathbf{u}_i, \mathbf{v}_j)$ of the deterministic model is simply the delta distribution. The key difference is that the Kullback-Leibler (KL) divergence [12] term in Equation 7 provides an additional soft constraint on the latent representation. This constraint encourages the high density area of the latent embedding distribution to be close to the origin (illustrated in Figure 3). This property enables the critiquing loop to be stable and encourages the post-critique latent embedding to remain in a domain compatible with the original embedding. We further explore these properties on real data in Section 3.4.1 and Figure 4.

During the critiquing stage, the probabilistic model produces updated rating and explanation predictions

$$\arg\max_{r_{i,j}, \mathbf{s}_{i,j}} E_{q(\mathbf{z_{i,j}}|\mathbf{u}_i, \mathbf{v}_j, \tilde{\mathbf{s}}_{i,j})}[\log p(r_{i,j}, \mathbf{s}_{i,j}, |\mathbf{z}_{i,j})], \tag{8}$$

by conditioning on the posterior latent representation distribution $q(\mathbf{z_{i,j}}|\mathbf{u}_i, \mathbf{v}_j, \tilde{\mathbf{s}}_{i,j})$. Specifically, the posterior distribution is a convenient closed-form product of two Gaussian distributions from the initial user-item embedding and the critiqued explanation inversion

$$\begin{aligned} & q(\mathbf{z_{i,j}}|\mathbf{u}_i, \mathbf{v}_j, \tilde{\mathbf{s}}_{i,j}) \\ &= \underbrace{q(\mathbf{z_{i,j}}|\mathbf{u}_i, \mathbf{v}_j)}_{\mathcal{N}(\mu_{\mathbf{u}_i,\mathbf{v}_j}, \sigma_{\mathbf{u}_i,\mathbf{v}_j})} \underbrace{p(\mathbf{z_{i,j}}|\tilde{\mathbf{s}}_{i,j})}_{\mathcal{N}(\mu_{\mathbf{s}_{i,j}}, \sigma_{\mathbf{s}_{i,j}})} \\ &= \mathcal{N}\Big(\frac{\sigma^2_{\mathbf{u}_i,\mathbf{v}_j}\mu_{\mathbf{u}_i,\mathbf{v}_j} + \sigma^2_{\mathbf{s}_{i,j}}\mu_{\mathbf{s}_{i,j}}}{\sigma^2_{\mathbf{u}_i,\mathbf{v}_j} + \sigma^2_{\mathbf{s}_{i,j}}}, \frac{\sigma^2_{\mathbf{u}_i,\mathbf{v}_j}\sigma^2_{\mathbf{s}_{i,j}}}{\sigma^2_{\mathbf{u}_i,\mathbf{v}_j} + \sigma^2_{\mathbf{s}_{i,j}}}\Big), \end{aligned} \tag{9}$$

where the posterior mean is a linear combination of the means of the original Gaussians weighted by their posterior variance.

Comparing the posterior latent believe update in Equation 9 and the deterministic latent belief update in Equation 4, we see that the former deterministic latent belief update is simply a special case of the latter posterior update, where specifically $\rho = \frac{\sigma^2_{\mathbf{u}_i,\mathbf{v}_j}}{\sigma^2_{\mathbf{u}_i,\mathbf{v}_j} + \sigma^2_{\mathbf{s}_{i,j}}}$.

Table 2: Summary of datasets. We selected 40 keyphrases for CDs&Vinyl and 75 keyphrases for BeerAdvocate. Coverage shows the percentage of reviews/comments that have at least one selected keyphrase.

Dataset	# Users	# Items	# Reviews	Sparsity	Keyphrase Coverage	Keyphrase Average Counts (per Review)
Beer (BeerAdvocate)	6,370	3,668	263,278	1.1268%	99.29%	7.7256
CDs&Vinyl (Amazon)	6,056	4,395	152,670	0.5736%	75.48%	2.1969

3 EXPERIMENTS

Now we proceed to evaluate the previously proposed CE-(V)NCF models in order to answer the following questions:

- Do the additional CE-(V)NCF training objectives for explanation and critiquing hurt recommendation performance compared to NCF and other state-of-the-art recommenders?
- Are CE-(V)NCF models able to produce reasonable explanations for their recommendations?
- Are CE-(V)NCF models effective on re-ranking recommendation by critiquing explanations?
- Which proposed model performs better, the variational probabilistic CE-VNCF model or the deterministic CE-NCF model?

All code to reproduce these results is publicly available on Github.[2]

3.1 Experiment Settings

3.1.1 Dataset. We evaluate the proposed CE-(V)NCF models on two publicly available datasets: BeerAdvocate [15] and Amazon CDs&Vinyl [9, 16]. Each of the datasets contains more than 100,000 reviews and product rating records. For the purpose of Top-N recommendation, we binarize the rating column of both datasets with a rating threshold τ. In CDs&Vinyl, the threshold is $\tau > 3$ out of 5. Due to the fact that people tend to rate positively in BeerAdvocate, we define the rating threshold $\tau > 4$ out of 5.

The datasets do not contain preselected keyphrases. Hence, we used the following generic processing steps to extract candidate keyphrases from the reviews to be used for explanation and critiquing for each dataset:

(1) Extract separate unigram and bigram lists of high frequency noun and adjective phrases from reviews of the entire dataset.
(2) Prune the bigram keyphrase list using a Pointwise Mutual Information (PMI) threshold to ensure bigrams are statistically unlikely to have occurred at random.
(3) Represent each review as a sparse 0-1 vector indicating whether each keyphrase occurred in the review.

While keyphrases are fixed, their usage for items is highly personalized. Table 2 shows overall dataset statistics for our experiments.

[2] https://github.com/wuga214/DeepCritiquingForRecSys

3.1.2 Evaluation Metrics. We evaluate the proposed framework from three quantitative perspectives: Top-N recommendation performance, Top-K explanation performance, and critiquing effectiveness. Metrics for each are discussed in the following subsections.

General Recommendation For Top-N recommendation performance, we compare the proposed methods with state-of-the-art recommenders on five metrics: MAP@N, Precision@N, Recall@N, R-Precision, and NDCG [11].

Explanation Generation For Top-K explanation generation performance, we report NDCG@K, MAP@K, Precision@K, and Recall@K for predicted keyphrases that a user would use to describe an item; results were evaluated on held-out test reviews that were not trained on. We compare our results with the ranked list of most popular keyphrases for each user (**UserPop**) and the most popular keyphrases for each item (**ItemPop**) to show that our explanation methods are both item-specific and personalized.

Critiquing Effectiveness Since the proposed CE-(V)NCF models are latent models where keyphrase explanations for (previously unreviewed) recommended items are personalized for a user, there is no explicit ground truth for the evaluation of critiquing. Therefore, we propose a novel evaluation metric called Falling MAP (F-MAP).

Given a set of items $\mathbb{S} = \{Item_j \mid j \in \{1 \cdots n\}\}$, and a critiquable keyphrase k, if k is in the Top-K explanation prediction of item j for user i, we say the item j belongs to the item set $\mathbb{S}^i_k$. Ideally, after user's i critique on k, we would want the rank of any affected items $\mathbb{S}^i_k$ to "fall" (move further down the ranked list) from the Top-N item recommendation list for user i after critiquing.

Using $\mathbb{S}^i_k$ as a surrogate to label ground truth "relevance" in a standard Mean Average Precision (MAP) metric, Falling MAP measures the ranking difference of the affected items set $\mathbb{S}^i_k$ before and after critiquing keyphrase k. Specifically,

$$F\text{-}MAP(i,k,N) = MAP@N^{\text{before}}_{\mathbb{S}^i_k} - MAP@N^{\text{after}}_{\mathbb{S}^i_k}, \qquad (10)$$

where N is the number of items to be recommended and $\mathbb{S}^i_k$ is cached before critiquing (i.e., keyphrases for items should not change after critiquing). We would expect the rank of items in $\mathbb{S}^i_k$ to fall after critiquing. *In summary, a positive F-MAP indicates that the critique had the intended effect on the latent embedding that negatively impacted both the rating and rank for the items most likely to have the critiqued explanation.* In our experiments, we average $F\text{-}MAP(i,k,N)$ over 1,000 user and keyphrase pairs.

3.1.3 Candidate Methods for General Performance Comparison. We compare general recommendation performance of the proposed framework with five state-of-the-art models including NCF:

- **POP:** Most popular items – not user personalized but an intuitive baseline to test the claims of this paper.
- **PureSVD [5]:** A similarity based recommendation method that constructs a similarity matrix through SVD decomposition of the implicit rating matrix.
- **CDAE [27]:** Collaborative Denoising Autoencoder – specifically optimized for implicit feedback recommendation tasks.
- **BPR [22]:** Bayesian Personalized Ranking – explicitly optimizes pairwise rankings.
- **NCF [10]:** Neural Collaborative Filtering. State-of-the-art deep learning based recommender.

Table 3: Best hyperparameter setting for each algorithm.

Domain	Algorithm	Optimizer	r	λ	Iteration*	α	β	η
Beer	CDAE	Adam	100	0.00001	300	0.0001	0.0	–
	BPR	Adam	50	0.00001	30	–	–	1
	PureSVD	–	100	1.0	10	–	–	–
	NCF	Adam	200	0.0001	300	0.001	–	5
	E-NCF	Adam	200	0.0001	300	0.001	–	5
	CE-NCF	Adam	200	0.0001	300	0.001	–	5
	VNCF	Adam	100	0.00005	300	0.001	0.1	5
	E-VNCF	Adam	100	0.00005	300	0.0005	0.1	5
	CE-VNCF	Adam	200	0.00005	300	0.001	0.1	5
CDs&Vinyl	CDAE	Adam	200	0.00001	300	0.0001	0.0	–
	BPR	Adam	200	0.0001	30	–	–	1
	PureSVD	–	200	1.0	10	–	–	–
	NCF	Adam	100	0.0001	300	0.0005	–	5
	E-NCF	Adam	100	0.001	300	0.0005	–	5
	CE-NCF	Adam	100	0.001	300	0.0005	–	5
	VNCF	Adam	200	0.00005	300	0.0001	0.1	5
	E-VNCF	Adam	200	0.00005	300	0.0001	0.1	5
	CE-VNCF	Adam	200	0.0001	300	0.0001	0.1	5

* For PureSVD, iterations in this table means the number of randomized SVD iterations. For BPR, CDAE, NCF, E-NCF, CD-NCF, VNCF, E-VNCF and CE-VNCF, iteration shows the number of epochs that processed over all users.

- **E-NCF:** Explainable NCF removes the critiquing loop from CE-NCF. This is one ablation of the CE-NCF model.
- **CE-NCF:** Full deterministic version of the proposed model.
- **VNCF:** Variational Extension of NCF model. This is one ablation of the CE-VNCF model that removes both explanation and critiquing.
- **E-VNCF:** Explainable-VNCF removes the critiquing loop from CE-VNCF. This is one ablation of the CE-VNCF model.
- **CE-VNCF:** Full variational probabilistic version of CE-NCF.

All NCF model architectures use: (1) Encoding network f_e: fully connected network (FCN) with ReLU activation. For all VNCF variants, the mean prediction passes through a ReLU activation, while the log-std prediction passes through a Tanh activation to restrict its range. (2) Rating prediction network f_r and key-phrase prediction network f_s: FCN with linear activation. (3) Inverse key-phrase prediction network $\tilde{f}_s^{-1}$: FCN with ReLU activation (same as f_e).

All experiments use 50% train, 20% validation, and 30% test data splits. Table 3 presents best hyperparameters tuned on validation data for each algorithm.[3] Corruption Rate β corresponds to a random zeroing of inputs $(\mathbf{u}_i, \mathbf{v}_j)$ inspired by CDAE [27]. Negative samples η corresponds to the number of uniformly randomly selected negative samples used for each positively sampled item.

3.2 Recommendation Performance

Before we evaluate the explanation and critiquing ability of the proposed models, we first need to confirm the proposed models achieve acceptable recommendation performance compared to state-of-the-art recommender systems.

Table 4 and 5 show the general recommendation performance comparison between the proposed models and various baselines. In the table, we make the following key observations:

[3] Hyperparameter tuning methodology is provided in the online Appendix.

Table 4: Results of Amazon CDs&Vinyl dataset. We omit the error bars since the confidence interval is in 4th digit.

Model	R-Precision	NDCG	MAP@5	MAP@10	MAP@20	Precision@5	Precision@10	Precision@20	Recall@5	Recall@10	Recall@20
POP	0.0078	0.0277	0.0096	0.0094	0.0088	0.0099	0.0087	0.0079	0.0088	0.0164	0.0317
CDAE	0.009	0.0313	0.0115	0.0113	0.0105	0.0115	0.0107	0.0094	0.0108	0.0206	0.0365
BPR	0.0621	0.1527	0.0719	0.0625	0.0524	0.0612	0.0489	0.0384	0.0751	0.116	0.1755
PureSVD	**0.0681**	0.1511	**0.078**	**0.0678**	**0.0559**	**0.0671**	0.0523	0.0389	**0.0846**	0.128	0.1821
NCF	0.06356	0.15688	0.07166	0.0634	0.05388	0.06272	0.05128	0.0399	0.07912	0.1253	0.18872
E-NCF	0.06442	0.1587	0.07218	0.06386	0.05436	0.06288	0.05162	0.04074	0.07922	0.12604	0.19084
CE-NCF	0.06266	0.15592	0.07058	0.06296	0.0536	0.06296	0.0512	0.0401	0.07806	0.12396	0.18724
VNCF	0.06716	0.16526	0.07398	0.06552	0.05576	0.06514	0.05304	0.04154	0.08326	0.13262	0.20102
E-VNCF	0.06736	**0.16532**	0.07394	0.06558	**0.0559**	0.06482	**0.05324**	**0.04168**	0.08328	**0.13358**	**0.20152**
CE-VNCF	0.06588	0.16388	0.07246	0.06468	0.05524	0.06398	0.053	0.04146	0.0811	0.13134	0.1996

Table 5: Results of Beer review dataset. We omit the error bars since the confidence interval is in 4th digit.

Model	R-Precision	NDCG	MAP@5	MAP@10	MAP@20	Precision@5	Precision@10	Precision@20	Recall@5	Recall@10	Recall@20
POP	0.0022	0.006	0.0027	0.0026	0.0028	0.0025	0.0024	0.0031	0.0009	0.0021	0.007
CDAE	0.0369	0.0886	0.0462	0.0427	0.0387	0.0424	0.0377	0.0328	0.0284	0.0491	0.0858
BPR	0.0349	0.0849	0.0421	0.039	0.0359	0.0379	0.035	0.0314	0.0258	0.0472	0.0839
PureSVD	0.0355	0.0812	0.0453	0.041	0.0365	0.0404	0.0346	0.0303	0.0279	0.0465	0.0788
NCF	0.0357	0.08723	0.04376	0.041	0.0377	0.04083	0.037	0.03266	0.02839	0.05173	0.08873
E-NCF	0.03643	0.08963	0.04609	0.04313	0.03933	0.04256	0.03863	0.0334	0.03106	0.0551	0.09299
CE-NCF	0.03726	0.0899	0.04656	0.04333	0.03946	0.04296	0.03866	0.03363	0.03143	0.05443	0.09153
VNCF	0.0436	0.10809	0.0539	**0.0508**	0.0466	**0.0509**	0.04576	0.04003	0.0367	0.0646	0.1099
E-VNCF	0.044	0.1084	**0.0542**	0.0507	0.04636	0.05036	0.0454	0.03973	**0.0368**	0.0647	0.1101
CE-VNCF	**0.0441**	**0.1084**	0.0538	0.0507	**0.0467**	0.0501	**0.0464**	**0.0402**	0.0358	**0.0651**	**0.1108**

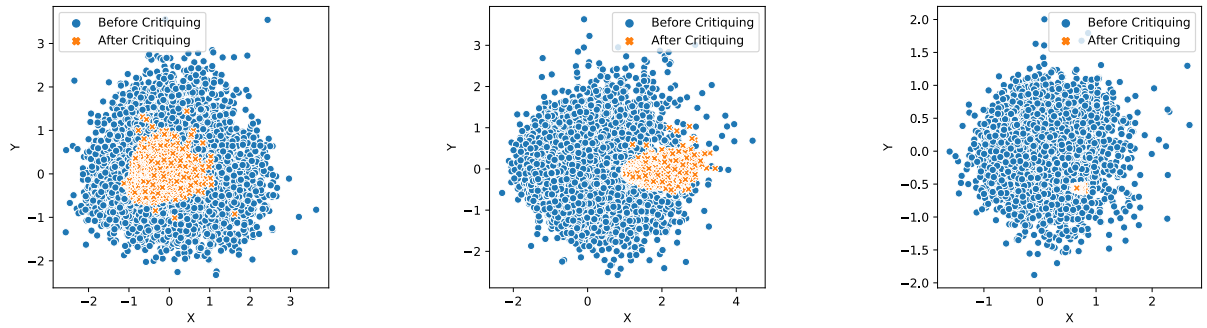
Table 6: Explanation Quality of CDs&Vinyl review dataset. We omit the error bars since the confidence interval is in 4th digit.

Model	NDCG@5	NDCG@10	NDCG@20	MAP@5	MAP@10	MAP@20	Precision@5	Precision@10	Precision@20	Recall@5	Recall@10	Recall@20
UserPop	0.1236	0.154	0.2225	0.1012	0.0886	0.0762	0.0932	0.069	0.065	0.1438	0.2118	0.4184
ItemPop	0.1385	0.1653	0.2313	0.108	0.0946	0.0797	0.1047	0.0716	0.065	0.1654	0.2259	0.4259
E-NCF	0.4717	0.5476	0.6101	0.359	0.2895	0.2199	0.2761	0.1925	0.1249	0.527	0.696	0.8729
CE-NCF	0.4736	0.5555	0.6186	0.3621	0.292	0.2226	0.2742	0.196	0.1266	0.5241	0.7066	0.885
E-VNCF	0.4797	0.5605	0.624	0.3668	0.2948	0.2245	0.2777	0.1968	0.1272	0.5291	0.7097	**0.8892**
CE-VNCF	**0.486**	**0.5662**	**0.627**	**0.3702**	**0.2986**	**0.227**	**0.2835**	**0.1995**	**0.1274**	**0.5397**	**0.7183**	**0.8892**

Table 7: Explanation Quality of Beer review dataset. We omit the error bars since the confidence interval is in 4th digit.

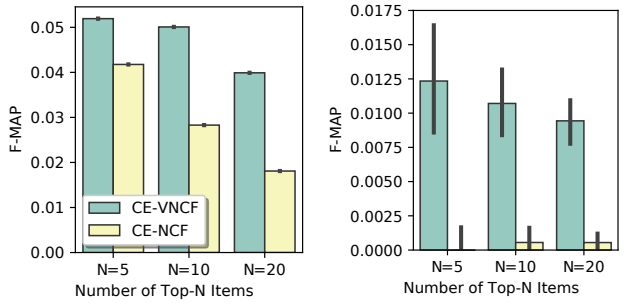
Model	NDCG@5	NDCG@10	NDCG@20	MAP@5	MAP@10	MAP@20	Precision@5	Precision@10	Precision@20	Recall@5	Recall@10	Recall@20
UserPop	0.0506	0.0809	0.1355	0.0707	0.0709	0.0739	0.0701	0.0744	0.0817	0.0428	0.0908	0.2026
ItemPop	0.0516	0.0805	0.1366	0.0743	0.0719	0.0731	0.0665	0.07	0.0806	0.0412	0.0868	0.202
E-NCF	0.30476	0.402	0.48456	0.42038	0.3774	0.31566	0.37838	0.3082	0.21954	0.25424	0.4096	0.57608
CE-NCF	0.29468	0.39048	0.47466	0.40684	0.36656	0.30838	0.36772	0.3014	0.21746	0.24586	0.39866	0.56854
E-VNCF	**0.3307**	**0.44362**	**0.53848**	**0.45468**	**0.41122**	**0.3478**	**0.4104**	**0.34266**	**0.24596**	**0.27696**	**0.45843**	**0.65022**
CE-VNCF	0.3181	0.42786	0.52282	0.4401	0.3981	0.33768	0.39698	0.33254	0.2415	0.26574	0.4417	0.63348

(1) Neural Collaborative Filtering (NCF), as the base model we build our proposed variants on, is a competitive recommendation algorithm in comparison to other state-of-the-art recommendation algorithms.
(2) Compared to the basic NCF model, our proposed variants show competitive performance in terms of all metrics, which shows that the additional training objectives for explanation and/or critiquing do not have a negative impact on the recommendation performance.
(3) The variational probabilistic inference models consistently outperform the deterministic models, showing the potential benefit of its KL-divergence regularizer, discussed shortly.
(4) For Amazon CD&Vinyl dataset, PureSVD outperforms all NCF based algorithms, which shows classic algorithms are often still competitive with state-of-the-art deep learning algorithms. Nonetheless NCF methods typically perform better on the Beer data set and the NCF architecture directly facilitates the latent critiquing models explored in this paper.



(a) CE-VNCF (b) CE-NCF Case 1 (c) CE-NCF Case 2

Figure 4: PCA 2D dimension reduction of latent representation z on the CDs&Vinyl dataset. Blue points show the distribution of the latent embedding *before* critiquing. Orange points show the distribution of the latent embedding *after* critiquing. (a) Shows a typical case for CE-VNCF training that projects the critiqued explanations back to the center of the same general latent space prior to critiquing. (b) Shows one training case where CE-NCF projects the critiqued explanations back to an off-center position from the general latent space prior to critiquing. (c) Shows a failure case of CE-NCF where the critiqued projection nearly collapses to a single point. We conjecture that the KL-Divergence regularization with an isotropic Gaussian for the latent z of CE-VNCF helps ensure critiqued projections remain origin-centered and helps prevent point mass collapse.



(a) Beer (b) CDs&Vinyl

Figure 5: Falling MAP versus Critiquing Models (higher is better). Error bars show the standard deviation. CE-NCF performs very poorly on CDs&Vinyl.

3.3 Explanation Prediction Performance

Table 6 and 7 show the keyphrase prediction performance comparison between the proposed models and popularity based baselines. The results show the proposed models outperform the user-wise and item-wise popularity prediction by more than 3 times on NDCG and MAP, and more than 2 times on Precision and Recall. Impressively, the proposed models are able to retrieve 88% percent of relevant keyphrases within the Top-20 explanations on CDs&Vinyl dataset that has around different 50 keyphrases.

Comparing to the critiquable models CE-(V)NCF and explainable models E-(V)NCF, we notice the explanation performance increased on CDs&Vinyl dataset but decreased on the Beer dataset. Considering the keyphrase sparsity of training data shown in Table 2, we conjecture that the latent reconstruction objective of critiquable models would help regularize the latent expression when keyphrase data is sparse, but *may* over-constrain learning when the keyphrase data is more dense.

3.4 Critiquing Performance

Figure 5 shows the F-MAP performance of our proposed critiquing models on both datasets. For the Beer dataset, we noticed that both CE-VNCF and CE-NCF show an anticipated positive F-MAP, where the rank of critiquing-affected items has dropped. CE-VNCF shows better performance over its deterministic simplified version on all three Top-N metrics, especially the case of N=20, where the performance of CE-VNCF doubles that of CE-NCF.

In contrast to the success of our models on the Beer dataset, we notice a higher variance in performance on the CDs&Vinyl dataset. Considering the statistical summary of this dataset in Table 2, we believe this variance stems from the higher sparsity of keyphrase observations, which makes it harder for each model to accurately predict personalized keyphrases. Figure 5(b) shows that CE-VNCF performs well as the rank of critique-affected items drops as expected. CE-NCF, on the other hand, completely fails on this dataset. The dramatic performance difference between the deterministic (CE-NCF) and variational probabilistic (CE-VNCF) models motivates us to better understand the reasons for these differences in the following section.

3.4.1 Latent Representation Analysis. Figure 4 illustrates the pre-and-post-critiquing latent representations in CE-VNCF and CE-NCF on the CDs&Vinyl dataset. The latent representations were dimensionally-reduced using PCA and the central region reflects an area of high density. Ideally, the latent codes should remain in the same general area to ensure that $f_r(\mathbf{z})$ and $f_s(\mathbf{z})$ remain valid.

Figure 4 (a) shows that for CE-VNCF, the latent representations after critiquing remain roughly in the same region as the original latent representations prior to critiquing. This observation is consistent with our hypothesis (Figure 3) that the KL divergence plays a crucial role in encouraging the reconstructed latent representations to remain in a valid region of latent space.

In contrast, Figure 4 (b) shows that the latent representation for CE-NCF is slightly shifted from the high density area of the original

Table 8: User Case Study on the Beer and CDs&Vinyl review datasets.

Dataset	Good Example?	User ID	Top Item Recommended	Initial Explanations	Critiqued Keyphrase	Refined Top Item Recommended	New Explanations
Beer	✓	672	Aecht Schlenkerla Rauchbier Urbock	Smoke, Bready, Brown	Smoke	Piraat Ale	Sweet, Fruit, Gold
	✓	433	∅lfabrikken Porter	Smooth, Chocolate, Black	Black	Brooklyn Brown Ale	Smooth, Brown, Sweet
	-	2794	10 Commandments	Honey, Sugar, Sweet	Sweet	Heady Topper	Fruit, Grapefruit, Orange
CDs&Vinyl	✓	3602	In The Zone	Pop, Dance, R&B	Dance	Under My Skin	Pop, Rock, Punk
	✓	828	Sgt. Pepper's Lonely Hearts Club Band	Rock, Pop, Ballad	Ballad	Life After Death	Rap, Hip Hop, Rock
	✗	2362	Rhythm Nation	R&B, Pop, Dance	Dance	Confessions	R&B, Pop, Techno

latent distribution. In this case, it would appear that hyperparameter tuning chose a weak regularizer that allowed the post-critique embedding to occupy a different space than the original embedding, leading to mismatched embeddings.

Even worse, Figure 4 (c) shows a complete failure case of CE-NCF, where the reconstructed latent representation after critiquing is nearly collapsed onto a single point. In this case, we observe in contrast that the CE-VNCF would be much less likely to let this happen since the KL-divergence regularization of CE-VNCF's latent space with an isotropic Gaussian prefers a non-collapsed latent distribution, an insight empirically illustrated by the broad distribution of the latent embedding of CE-VNCF in Figure 4 (a).

In summary, our investigation of the latent space distribution of pre- and post-critiqued embeddings suggests that the KL-Divergence regularization with an isotropic Gaussian of CE-VNCF helps prevent latent space pathologies of skewed or collapsed post-critiqued embeddings that appear evident in CE-NCF. And these potential pathologies – effectively incompatibility between the pre- and post-critiqued latent embeddings – would then help explain the relatively poor critiquing performance evaluation of CE-NCF compared to CE-VNCF observed in Figure 5.

3.5 Case Study

To qualitatively evaluate the performance of the critiquing in a real environment, we simulated multiple use-cases of the proposed model (CE-VNCF) on the two review datasets. Table 8 shows six representative examples we encountered during our investigation. The examples were manually categorized as good/bad examples based on analyzing product details and reviews on the BeerAdvocate and Amazon websites.

For the Beer dataset, we see in the first example that the model recommends *Aecht Schlenkerla Rauchbier Urbock* to user 672 with keyphrase explanations: *Smoke*, *Bready* and *Brown*. The BeerAdvocate website lists the following comment for *Aecht Schlenkerla Rauchbier Urbock* : *"Deep brown in color with some light seeping through when held up to the light. There's an inch of creamy and dense beige foam on top. . . . I'm tasting that sweet, dark malt and then the smoked flavor kicks in and lingers after the finish. . . . "*, which demonstrates that the generated explanation accurately reflects the beer's properties. We then chose to critique the keyphrase *Smoke*, resulting in a refined top recommendation with attributes *Sweet* and *Fruit*, which are contrasting tastes to *Smoke*. In the second example, we critiqued the color *Black* and the refined recommended beer has a different color *Brown*, but the same *Smooth* mouthfeel of the initial recommendation.

Turning our attention to the CDs&Vinyl dataset, in the first example, the model recommends the album *In the Zone*. After critiquing *Dance*, the model recommends *Under My Skin*, which is primarily a *Pop* album without the *Dance* attribute. In the second example, critiquing the keyphrase *Ballad* yields *Life After Death*, which is not a *Ballad* but remains a *Rock* album; one of the guest artists of *Life After Death* is DMC who produces rap rock.

The above anecdotal examples demonstrate that critiquing with CE-VNCF can yield reasonable refined recommendations. Although general performance was good on the two datasets, we observed several unsatisfactory cases exemplified by the last examples in Table 7. For CDs&Vinyl, even though the *Dance* keyphrase was critiqued, the refined recommendation was Usher's album *Confessions* which is *Techno*, a form of electronic dance music. Here, the model failed to infer that *Techno* is a subclass of *Dance*, which is a taxonomic inference not directly supported by our latent model. We see a similar effect in the third example of the Beer dataset; all of the top-three keyphrases suggest that the original recommendation *10 Commandments* is a sweet-tasting beer. The only critiqued target is *Sweet* and none of the original keyphrases are preserved in the refined recommendation. Nevertheless, *Heady Topper* possesses similar keyphrases and has the following online review: *"A sweet and fruity IPA with a slight bitterness at the finish."*. An interesting future direction is to leverage latent explanation and critiquing methods that can reason about implications between keyphrases.

4 CONCLUSION

In this paper, we proposed two novel end-to-end deep learning frameworks – one deterministic and one probabilistic – that extended Neural Collaborative Filtering (NCF) [10] recommendation with explanation and critiquing components. On two datasets, we observed that both frameworks provide strong recommendation performance and high-quality personalized item keyphrase suggestions, but that the variational probabilistic method CE-VNCF performs best on the critiquing evaluation. To explain this latter result, we analyzed how the KL-divergence regularizer of CE-VNCF yielded the most compatible co-embeddings of user and item preferences with language-based critiques. Our framework can be readily applied to various domains and extended to other collaborative interactive settings such as group recommendation and item generation [25]. Overall, we hope this work provides a rich foundation for future extensions of deep language-based critiquing in conversational recommender systems, e.g., leveraging more complex deep explanation and language-based feedback structure as well as multi-step sequential interactions and active learning methods.

REFERENCES

[1] Dana H Ballard. 1987. Modular learning in neural networks. In *Proceedings of the sixth National conference on Artificial intelligence-Volume 1*. AAAI Press, 279–284.

[2] Robin D. Burke, Kristian J. Hammond, and Benjamin C. Young. 1996. Knowledge-based Navigation of Complex Information Spaces. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 1 (AAAI'96)*. AAAI Press, 462–468.

[3] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 815–824.

[4] Felipe Costa, Sixun Ouyang, Peter Dolog, and Aonghus Lawlor. 2018. Automatic generation of natural language explanations. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion*. ACM, 57.

[5] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 39–46.

[6] N Benjamin Erichson, Sergey Voronin, Steven L Brunton, and J Nathan Kutz. 2016. Randomized matrix decompositions using R. *arXiv preprint arXiv:1608.02148* (2016).

[7] A. Felfernig and R. Burke. 2008. Constraint-based Recommender Systems: Technologies and Research Issues. In *Proceedings of the 10th International Conference on Electronic Commerce (ICEC '08)*. ACM, New York, NY, USA, Article 3, 3:1–3:10 pages.

[8] Peter Grasch, Alexander Felfernig, and Florian Reinfrank. 2013. ReComment: Towards Critiquing-based Recommendation with Speech Interaction. In *Proceedings of the 7th ACM Conference on Recommender Systems (RECSYS)-13*. New York, NY, USA, 157–164.

[9] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 507–517.

[10] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 173–182.

[11] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[12] Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics* 22, 1 (1951), 79–86.

[13] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. 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*. ACM, 345–354.

[14] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *Proceedings of the 2018 World Wide Web Conference (WWW '18)*. Republic and Canton of Geneva, Switzerland, 689–698.

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

[16] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 43–52.

[17] Kevin McCarthy, Yasser Salem, and Barry Smyth. 2010. Experience-based critiquing: Reusing critiquing experiences to improve conversational recommendation. In *International Conference on Case-Based Reasoning*. Springer, 480–494.

[18] Jennifer Nguyen and Mu Zhu. 2013. Content-boosted matrix factorization techniques for recommender systems. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 6, 4 (2013), 286–301.

[19] James Reilly, Kevin McCarthy, Lorraine Mcginty, and Barry Smyth. 2004. Dynamic Critiquing. In *Advances in Case-Based Reasoning, 7th European Conference (ECCBR) 2004*. 37–50. https://doi.org/10.1007/978-3-540-28631-8_55

[20] James Reilly, Kevin McCarthy, Lorraine McGinty, and Barry Smyth. 2004. Incremental critiquing. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence*. Springer, 101–114.

[21] James Reilly, Kevin McCarthy, Lorraine McGinty, and Barry Smyth. 2005. Explaining Compound Critiques. *Artif. Intell. Rev.* 24, 2 (Oct. 2005), 199–220.

[22] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 452–461.

[23] S. Sedhain, A. Menon, S. Sanner, and L. Xie. 2015. AutoRec: Autoencoders Meet Collaborative Filtering. In *Proceedings of the 24th International Conference on the World Wide Web (WWW-15)*. Florence, Italy.

[24] Cynthia A Thompson, Mehmet H Goker, and Pat Langley. 2004. A personalized system for conversational recommendations. *Journal of Artificial Intelligence Research* 21 (2004), 393–428.

[25] Thanh Vinh Vo and Harold Soh. 2018. Generation Meets Recommendation: Proposing Novel Items for Groups of Users. In *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18)*. ACM, New York, NY, USA, 145–153. https://doi.org/10.1145/3240323.3240357

[26] Ga Wu, Maksims Volkovs, Chee Loong Soon, Scott Sanner, and Himanshu Rai. 2019. Noise Contrastive Estimation for Scalable Linear Models for One-Class Collaborative Filtering. *The 42th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[27] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, 153–162.

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

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

[30] Xiaoxue Zhao, Weinan Zhang, and Jun Wang. 2013. Interactive collaborative filtering. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, 1411–1420.