Dual Learning for Explainable Recommendation: Towards Unifying User Preference Prediction and Review Generation

Peijie Sun
Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology
School of Computer Science and Information Engineering, HeFei University of Technology
sun.hfut@gmail.com

Le Wu
Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology
School of Computer Science and Information Engineering, HeFei University of Technology
lewu.ustc@gmail.com

Kun Zhang
School of Computer Science and Technology, University of Science and Technology of China
zhkun@mail.ustc.edu.cn

Yanjie Fu
University of Central Florida
yanjie.fu@ucf.edu

Richang Hong
Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology
School of Computer Science and Information Engineering, HeFei University of Technology
hongrc.hfut@gmail.com

Meng Wang*
Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology
School of Computer Science and Information Engineering, HeFei University of Technology
eric.mengwang@gmail.com

ABSTRACT

In many recommender systems, users express item opinions through two kinds of behaviors: giving preferences and writing detailed reviews. As both kinds of behaviors reflect users' assessment of items, review enhanced recommender systems leverage these two kinds of user behaviors to boost recommendation performance. On the one hand, researchers proposed to better model the user and item embeddings with additional review information for enhancing preference prediction accuracy. On the other hand, some recent works focused on automatically generating item reviews for recommendation explanations with related user and item embeddings. We argue that, while the task of preference prediction with the accuracy goal is well recognized in the community, the task of generating reviews for explainable recommendation is also important to gain user trust and increase conversion rate. Some preliminary attempts have considered jointly modeling these two tasks, with the user and item embeddings are shared. These studies empirically showed that these two tasks are correlated, and jointly modeling them would benefit the performance of both tasks.

In this paper, we make a further study of unifying these two tasks for explainable recommendation. Instead of simply correlating these two tasks with shared user and item embeddings, we argue that these two tasks are presented in dual forms. In other words, the input of the primal preference prediction task $p(\mathcal{R}|C)$ is exactly the output of the dual review generation task $p(C|\mathcal{R})$, with $\mathcal{R}$ and C denote the preference value space and review space. Therefore, we could explicitly model the probabilistic correlation between these two dual tasks with $p(\mathcal{R},C)$=$p(\mathcal{R}|C)p(C)$=$p(C|\mathcal{R})p(\mathcal{R})$. We design a unified dual framework of how to inject the probabilistic duality of the two tasks in the training stage. Furthermore, as the detailed preference and review information are not available for each user-item pair in the test stage, we propose a transfer learning based model for preference prediction and review generation. Finally, extensive experimental results on two real-world datasets clearly show the effectiveness of our proposed model for both user preference prediction and review generation.

CCS CONCEPTS

• **Information systems** → **Collaborative filtering**; • **Computing methodologies** → *Natural language generation*.

KEYWORDS

recommender system, dual learning, review generation

ACM Reference Format:
Peijie Sun, Le Wu, Kun Zhang, Yanjie Fu, Richang Hong, and Meng Wang. 2020. Dual Learning for Explainable Recommendation: Towards Unifying User Preference Prediction and Review Generation . In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3366423.3380164

1 INTRODUCTION

Collaborative Filtering (CF) based recommendation provides personalized item suggestions to users based on their historical preference, and has been widely studied in the past with high recommendation accuracy [11, 21, 24]. However, CF suffers from the cold-start problem. Therefore, many auxiliary data, such as review information [4, 30, 45], social networks [5, 37], and knowledge graph [32, 33], are leveraged with CF to enhance recommendation

*Corresponding author.

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380164

performance. Among them, reviews are very common in most platforms, which express the users' detailed feelings to the items with rich semantic information. In essence, writing reviews and giving preferences are two facets of showing users' opinions to items. Therefore, review enhanced recommender systems are studied for better recommendation with the consideration of these two kinds of users' behaviors [14, 30].

Currently, given the user preference space $\mathcal{R}$ and review space C, the solutions for review enhanced recommendation can be classified into two categories. First, to alleviate the data sparsity in CF, researchers proposed to take reviews as input for better user and item embedding learning, such that the preference prediction task could be improved [4, 30, 45]. These models predict the conditional distribution $p(\mathcal{R}|C;\theta)$, with θ is the parameter set that contains user and item embeddings, as well as other parameters. For example, as reviews are concerned with both users and items, DeepCoNN learned the hidden user and item representations with two parallel CNN based models [45]. In summary, this line of research focuses on how to inject the review semantic information into CF, and greatly improves recommendation accuracy.

Second, reviews are also widely used as training data to train an explainable recommendation model which can provide semantic explanations. Explanations would help to gain trust between users and the systems, and encourage users to make decisions [12, 35, 36, 43]. While some researchers proposed to identify and extract detailed sentiments for recommendation explanation [45], these models could not tackle the situation when an item has very few reviews. In the meantime, neural network based language generation models have shown state-of-the-art performance in many natural language generation areas, such as language translation [29], image caption [39] and text representation [41]. As such, a more natural choice is to automatically generate reviews for explainable recommendations [7, 30]. Attribute to sequence models have been proposed for producing review generation, with the attributes are associated with user and item latent embeddings, user profiles and item features [13, 46]. These models predicted the conditional distribution $p(C|\mathcal{R};\varphi)$, with φ is the parameter set that contains user and item embeddings, as well as other model parameters.

In fact, the two tasks of predicting user preference for item recommendation and generating reviews for the recommended items are not isolated but correlated. Both tasks rely on the related user's and item's embeddings based on the same candidate user-item pair. Based on this intuition, there are some recent works that jointly modeled these two tasks toward accurate and explainable recommendation [14, 30]. Specifically, the user embeddings and item embeddings are shared among the two tasks. Then, a joint optimization function is proposed to linearly combine the loss of preference prediction and language generation. Different models vary in the detailed formulation of the two conditional likelihood of the two tasks (i.e., $p(\mathcal{R}|C)$, $p(C|\mathcal{R})$). These models have shown superior performance by considering these two tasks in a unified model, empirically showing the mutual reinforcement relationship between these two tasks.

In this paper, we further study the jointly modeling problem. Instead of simply correlating these two tasks with shared parameters, we argue that these two tasks are presented in *dual forms*, and have intrinsic probabilistic connections between them. Specifically, in review based recommender systems, with user preference prediction as $p(\mathcal{R}|C;\theta)$, and review generation as $p(C|\mathcal{R};\varphi)$, we find the input of either task is exactly the output of the remaining task. Dual learning has emerged in many real-world machine learning scenarios, such as image translation, machine translation, and image classification and generation [8, 38].

Dual learning based theories are designed to tackle the correlation between dual tasks with probabilistic correlation. By treating the two tasks in review enhanced recommendation as dual tasks, the following probabilistic correlation are better satisfied: $p(\mathcal{R},C) = p(\mathcal{R}|C)p(C) = p(C|\mathcal{R})p(\mathcal{R})$. Therefore, in this paper, we would like to correlate these two tasks under the dual learning framework for jointly predicting user preference and generating review explanations. In practice, we turn the probabilistic duality as a constraint for correlating these two tasks. Without the dual learning framework, previous works optimized the correlation of the two tasks with shared parameters, and could not guarantee the probabilistic duality constraint. In contrast, by explicitly modeling the duality constraint, the two tasks are explicitly correlated, and the learning process of the two tasks could be strengthened with each other, pushing the learning of the two tasks towards the right direction.

However, dual modeling of the two tasks for recommendation is non-trivial due to the following two challenges: First, in the training process, how to push the duality forms of the two tasks in the modeling process? Second, in the test stage, for each candidate user-item pair, we do not have either the preference information or the review data. How to make recommendations in the test stage when the input data is not available? To tackle these two challenges, we first propose a unified dual model to inject the probabilistic duality of the two tasks in the training stage, with each probability function is well designed. Furthermore, to tackle the unavailable preference and review in the test stage, we propose a transfer learning based model to simulate the review representation of each user-item pair, which could be approximated for preference prediction and review generation. In summary, the main contribution of this paper is summarized as follows:

- In this paper, we argue that two tasks of user preference prediction and review generation are naturally presented in dual forms. By forming the duality between them, we propose to jointly model these two tasks with with probabilistic dual correlation.
- We propose a dual learning based joint model for user preference prediction and review generation. Our main technical contribution lies in designing detailed probability modeling of each task in the training process with duality constraints. Besides, we design a transfer learning based model in the test stage to solve the unavailable preference and review problem.
- We conduct extensive experiments on two real-world datasets. The experimental results clearly show the effectiveness of our proposed model for user preference prediction and review generation compared to state-of-the-art models.

2 RELATED WORK

Traditionally, CF models relied on user-item historical behavior for learning user and item representations in a low collaborative latent space, and received success in the past [11, 21, 24]. E.g., Probabilistic Matrix Factorization (PMF) adopted matrix factorization to achieve this goal [21]. However, CF suffered from the data sparsity issue, as many users have a few historical records. Reviews are common in most recommender systems and have been widely applied to enhance recommendation accuracy [1, 4, 17, 19, 26, 45]. Some earlier works focused on how to boost the performance by treating reviews as side information, such as modeling items' preferences with LDA topic model[1], or regularizing the matrix factorization training process with reviews[17], or treating the reviews as features with Factorization Machines [23]. With the huge success of deep learning in NLP, researchers proposed to align better review content embedding into the user and item embedding modeling process [4]. DeepCoNN is a state-of-the-art deep learning based recommendation model with two parallel CNN based content embedding modules, with the reviews associated to the users and items as input [45]. Different models varied in the detailed choices of the state-of-the-art text embedding methods [18, 25], and the design of attention networks for selecting important semantic information [4, 26]. Researchers have found for each user-item pair, the review from the user to the item is the most valuable for predicting preferences. However, the detailed review is not available in the test stage. TransNet is proposed to approximate the review representation of the user-item pair, and showed better performance in practice [3].

Besides utilizing reviews to enhance recommendation accuracy, reviews have also been applied for explainable recommendation. Some works focused on extraction techniques, such as modeling item's and users' embeddings based on the aspects [9, 43], selecting valuable sentences for recommendation with either attention modeling [4] or reinforcement modeling techniques [34]. While directly using extraction based models may suffer from the copyright related issues and limited reviews, a more popular approach is to borrow the advances in natural language generation to generate reviews for explainable recommendation [2, 29]. The state-of-the-art models for natural language generation models follow an encoder-decoder structure, with encoding all related information, and decoding a sentence. Natural language generation has been widely studied in domains such as machine translation [2], image captioning [39], dialogue systems [27], and so on. Researchers proposed an attribute to sequence model to learn to generate reviews for products, where the encoder part includes users' profiles, items' attributes and users' preferences to the items [46]. Researchers also leveraged auxiliary sources to enhance text generation performance, e.g., the auxiliary item information associated with items [42, 44], user persona language styles [13]. In summary, these models either assumed the user related content or the user embedding is available for the generation tasks, and suffered from two limitations for recommendation. First, these generation tasks failed for the platforms that do not have any user profile. Second, the user embedding is usually learned from users' preferences, and separating the task of user preference learning and review generation would lead to suboptimal performance.

Recently, there are several attempts that proposed to jointly learn user preferences and generate reviews with a multi-task learning framework [7, 14, 16, 30]. The key ideas of these models are composed of two tasks: a user preference learning task with user and item embeddings as input, and a review generation task that also takes user and item embeddings in the encoder. Therefore, the user and item embeddings are shared among these two tasks. E.g., researchers proposed a Multimodal Review Generation (MRG) model for these two tasks with shared user and item latent embeddings[30]. Besides, the multimodal information of items is encoded for better review generation. After that, a joint optimization function with shared embeddings is proposed to combine both preference prediction error and review generation error for multi-task learning [30]. Different models varied in the detailed implementations with auxiliary data, i.e., the tips information that are abstractions of reviews [14], the associated knowledge base information of items [30], or the fine-grained image semantics that are specifically designed for fashion recommendation [6, 16]. We differ greatly from these works as we do not put emphasis on how to design more sophisticated models for either preference prediction and review generation, and our main contribution lies in correlating these two tasks under a dual form, such that the probabilistic correlation of these two tasks are modeled in the multi-task learning process. In fact, our proposed dual learning framework is flexible and could be easily extended to any of the above models with consideration of duality constraints.

Our work is also closely related to the theories in dual learning. Many machine learning tasks present in dual forms, such as duality in machine translation between two languages, image caption task and image generation task in image processing. Dual learning leverages the symmetric (primal-dual) structure of two tasks to enhance learning process [8]. Duality learning can be presented in the unsupervised setting [8], and the supervised setting [38]. For dual unsupervised learning, we could only observe the duality between two domains without the exact each pair correlation, such as unpaired image translation [47] and unpaired machine translation [38]. With one model for the primal task and another model for the dual task, these two tasks would teach each other for mutual reinforcement. Our work is more closely related to the dual supervised learning [38], with the correlation between each preference and review pair is available. In dual supervised learning, the correlation between two tasks is modeled with dual probabilistic correlation, and usually is turned to a data dependent regularization term for optimization. Please refer to the original paper for more details [38]. To the best of our knowledge, we are one of the first few attempts that leverage the duality between user preference prediction and review generation for better learning of the two tasks.

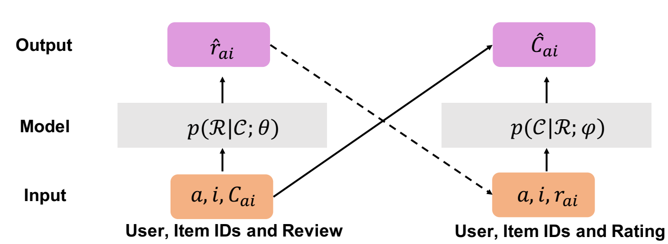


Figure 1: An illustration of the duality between the preference prediction task and review generation task.

3 THE PROPOSED FRAMEWORK

In this section, we design *DualPC*: a *Dual* learning based framework with both *P*reference prediction and review *C*ontent generation in review based recommender systems. We would first give the overall framework of DualPC, followed by the submodules in the framework. After that, we describe the training process of DualPC, and how to test our model when the candidate user-item preference and review content are not available.

3.1 Overall Framework of DualPC

In a review based recommender system, the training set is $\mathcal{X} = (\mathcal{U}, \mathcal{V}, \mathcal{R}, C)$, where $\mathcal{U}$ and $\mathcal{V}$ are the user set and item set respectively. $\mathcal{R}$ is the preference set, with each preference value can be explicit numerical preferences or implicit feedback. C is the review set, with each review is a sentence written by a user to an item. Given the training set $\mathcal{X}$, we have a preference prediction task $p(\mathcal{R}|C;\theta)$ that predicts user preference based on the reviews with parameters θ, and a review generation task $p(C|\mathcal{R};\varphi)$ that takes user-item related preference as input for review generation with parameters φ.

Given each training record of as a quad of x=(a, i, r_{ai}, C_{ai}), in which a is a user and i is an item, r_{ai} and C_{ai} are the preference and review of a to i. The preference prediction task aims to find a function f: $a, i, C_{ai} \rightarrow r_{ai}$, which is used to maximize the conditional probability $p(r_{ai}|a, i, C_{ai};\theta)$ of the real preference r_{ai}. As for the review generation task, it aims to find a function g: $a, i, r_{ai} \rightarrow C_{ai}$, which is used to maximize the conditional probability $p(C_{ai}|a, i, r_{ai};\varphi)$ of the real review C_{ai}. These two tasks can be formulated as follows:

$$f(a,\, i,\, C_{ai};\theta) \triangleq \arg\max p(\mathcal{R}|C;\theta) = \arg\max \prod_{x\in\mathcal{X}} p(r_{ai}|a,\, i,\, C_{ai};\theta),$$
$$g(a,\, i,\, r_{ai};\varphi) \triangleq \arg\max p(C|\mathcal{R};\varphi) = \arg\max \prod_{x\in\mathcal{X}} p(C_{ai}|a,\, i,\, r_{ai};\varphi)]$$

where θ and φ are the trainable parameter sets of preference prediction model f and review generation model g. Please note that, without confusion, we would omit the term of (a, i) (e.g., $p(r_{ai}|a, i, C_{ai})$ is denoted as $p(r_{ai}|C_{ai})$).

By many standard learning tasks, the preference prediction model f is learned by minimizing the empirical risk between predicted preference and the training data as:

$$\min_{\theta} \frac{1}{|\mathcal{X}|} \sum_{x\in\mathcal{X}} l_1(f(a,\, i,\, C_{ai};\theta),\, r_{ai}); \tag{1}$$

and the review content generation model g is similarly learned by:

$$\min_{\varphi} \frac{1}{|\mathcal{X}|} \sum_{x\in\mathcal{X}} l_2(g(a,\, i,\, r_{ai};\varphi),\, C_{ai}). \tag{2}$$

In fact, as shown in Fig.1, these two tasks are presented in dual forms. Therefore, we associate these two tasks with dual learning. In the following, we treat the preference prediction task as the primal task and review content generation as the dual task. If the learned primal and dual models are perfect, the probabilistic duality between them are better satisfied:

$$\begin{aligned} p(\mathcal{X}) &= \prod_{x\in\mathcal{X}} p(a,\, i,\, r_{ai},\, C_{ai}) \\ &= \prod_{x\in\mathcal{X}} p(C_{ai})p(r_{ai}|C_{ai};\theta) = \prod_{x\in\mathcal{X}} p(r_{ai})p(C_{ai}|r_{ai};\varphi) \end{aligned} \tag{3}$$

By combining the probabilistic duality correlation between the two tasks in Eq.3 with the empirical losses of the two tasks (i.e., Eq.1 and Eq.2), we optimize the following objective optimization function as:

$$\begin{aligned} &objective1: \min_{\theta} \frac{1}{|\mathcal{X}|} \sum_{x\in\mathcal{X}} l_1(f(a,\, i,\, C_{ai};\theta),\, r_{ai}), \\ &objective2: \min_{\varphi} \frac{1}{|\mathcal{X}|} \sum_{x\in\mathcal{X}} l_2(g(a,\, i,\, r_{ai};\varphi),\, C_{ai}), \\ &\text{s.t.} \quad \prod_{x\in\mathcal{X}} p(C_{ai})p(r_{ai}|C_{ai};\theta) = \prod_{x\in\mathcal{X}} p(r_{ai})p(C_{ai}|r_{ai};\varphi) \end{aligned}$$

where $p(r_{ai})$ and $p(C_{ai})$ are the marginal distributions.

As directly putting the probabilistic constraints (Eq.3) into the two objectives are not feasible in practice, we convert the duality correlation as a regularization term:

$$\begin{aligned} l_{duality} = &\sum_{x\in\mathcal{X}} [logp(r_{ai}) + logp(C_{ai}|r_{ai};\varphi) \\ &- logp(C_{ai}) - logp(r_{ai}|C_{ai};\theta)]^2. \end{aligned} \tag{4}$$

Therefore, the probabilistic correlation of the two tasks are modeled by the above duality based regularization term. Under the DualPC framework, for each task, we could reformulate the optimization goal by the weighted combination between the original loss function and the above duality based regularization term:

$$\text{objective1:} \min_{\theta} \frac{1}{|\mathcal{X}|} \sum_{x\in\mathcal{X}} (l_1 + \lambda_1 l_{duality}), \tag{5}$$

$$\text{objective2:} \min_{\varphi} \frac{1}{|\mathcal{X}|} \sum_{x\in\mathcal{X}} (l_2 + \lambda_2 l_{duality}), \tag{6}$$

where λ_1 and λ_2 are the weights of the duality terms that need to be tuned in practice. They control the gradient from the duality loss when optimizing θ and ϕ respectively. With larger values of them, the duality regularization terms play more important roles in correlating these two tasks with probabilistic constraints.

However, the objective functions are different from the following one:

$$\min_{\theta} \frac{1}{|\mathcal{X}|} \sum_{x\in\mathcal{X}} (l_1 + l_2 + (\lambda_1 + \lambda_2) l_{duality})$$

as the coefficient of the gradient from the duality loss term is a fixed value (λ_1+λ_2).

In the following, we would like to show the sub modules of DualPC framework with preference prediction (Eq.5) and review generation (Eq.6). Specifically, we would first show how to compute the original losses introduced by the preference prediction function f (Eq.1) and the review generation function g (Eq.2). After that, we model the duality loss (Eq.4) by introducing the calculation of the marginal distributions. We then give the model training process.

3.2 Preference Prediction Module of DualPC

In this subsection, we design a submodule for DualPC for the primary preference prediction task given the review data. In other words,we model the predicted preference of $\hat{r}_{ai}$ as: $\hat{r}_{ai}$=$f(a, i, C_{ai};\theta)$ with parameter set θ (Eq.1). We show the structure of the preference prediction in Fig.2, which includes the embedding layers, the fusion layer, the Multi-layer Perceptions (MLPs).

Embedding Layer. Each input record is associated with a user ID a, an item ID i, and review C_{ai} that is consists of T words $(C^0_{ai}, C^1_{ai}, C^2_{ai}, ..., C^t_{ai}, ..., C^{T-1}_{ai})$. Similar as many CF based models,

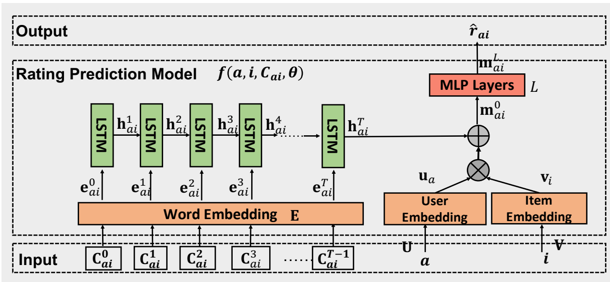


Figure 2: The overall structure of the preference prediction module with review input: $f(a, i, C_{ai}; \theta)$**.**

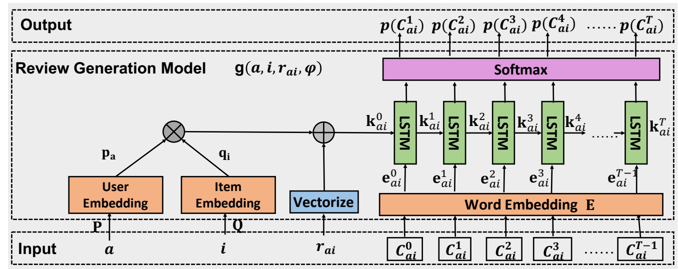


Figure 3: The overall structure of the review generation module with preference input: $g(a, i, r_{ai}; \varphi)$**.**

we first turn the user ID and item ID into embeddings with free user embedding matrix $\mathbf{U} \in \mathbb{R}^{D\times|\mathcal{U}|}$ and free item embedding matrix $\mathbf{V} \in \mathbb{R}^{D\times|\mathcal{V}|}$. Therefore, user a's embedding vector is the a^{th} column of $\mathbf{U}$, denoted as $\mathbf{u}_a$. Similarly, item i's embedding vector $\mathbf{v}_i$ is the i^{th} column of $\mathbf{V}$.

For reviews, since the focus of this paper is not to design more sophisticated models for review embedding, we select the pre-trained LSTM to get review embedding[28], which has been widely used in many language modeling tasks with state-of-the-art performance. Thus, each review can be represented with the pre-trained LSTM as follows:

$$\mathbf{h}_{ai}^{T} = LSTM(C_{ai}), \tag{7}$$

Specifically, at each time t, the hidden state of the LSTM structure is $\mathbf{h}_{ai}^{t} = LSTM(\mathbf{h}_{ai}^{t-1}, \mathbf{e}_{ai}^{t-1}; \theta_{LSTM})$, with θ_{LSTM} is the trainable parameters set of the LSTM structure. $\mathbf{e}_{ai}^{t-1}$ is the word embedding of the $(t\text{-}1)^{th}$ word in C_{ai}. In fact, each word embedding could be pretrained from the word embedding matrix $\mathbf{E}$. We treat the last hidden state $\mathbf{h}_{ai}^{T}$ of the LSTM as the representation of the input review (T is the number of words in the review).

Fusion Layer. The fusion layer fuses the input embeddings as:

$$\mathbf{m}_{ai}^{0} = [\mathbf{h}_{ai}^{T}, \mathbf{u}_a \circ \mathbf{v}_i], \tag{8}$$

where $\mathbf{u}_a \circ \mathbf{v}_i$ is the element-wise product that models the collaborative interaction between each user-item pair.

MLP Layers. By feeding the output of the fusion layer into MLP layers, this layer-wise structure models the complex interactions between the collaborative signals and review semantics. Suppose there are L layers of MLPs, we have

$$\mathbf{m}_{ai}^{l} = ReLU(\mathbf{W}^{(l-1)} \times \mathbf{m}_{ai}^{(l-1)} + \mathbf{b}^{l-1}), \, l = 1, \ldots, L. \tag{9}$$

Preference Prediction Layer. By feeding the output of MLPs, i.e., $\mathbf{m}_{ai}^{L}$, into the preference layer, we have:

$$\hat{r}_{ai} = \mathbf{w}' \times \mathbf{m}_{ai}^{L} + b, \tag{10}$$

where $\mathbf{w}'$ is the vector parameter, and b is a bias term that needs to be learned.

After obtaining the predicted preferences, similar as many preference prediction tasks, we assume the likelihood of each preference record r_{ai} follow a Gaussian distribution, with the mean of the predicted preference $\hat{r}_{ai}$ and variance of σ^2 [21]. Then, the likelihood of the conditional preference distributions is modeled as:

$$p(R|C; \theta) = \prod_{x\in\mathcal{X}} \mathrm{N}(r_{ai}|\hat{r}_{ai}, \sigma^2) = \prod_{x\in\mathcal{X}} \frac{1}{\sqrt{2\pi\sigma^2}} exp(\frac{-(r_{ai} - \hat{r}_{ai})^2}{2\sigma^2}). \tag{11}$$

where σ^2 is a hyperparameter that needs to be tuned.

Given the above likelihood function, maximizing the above conditional likelihood function is equivalent to minimizing the log-likelihood optimization function (Eq.1) as:

$$\begin{aligned} &\min_{\theta} \frac{1}{|\mathcal{X}|} \sum_{x\in\mathcal{X}} l_1(f(a, i, C_{ai}; \theta), r_{ai}) \\ =&\min_{\theta} \frac{1}{|\mathcal{X}|} \sum_{x\in\mathcal{X}} log(\sqrt{2\pi\sigma^2}) + \frac{(r_{ai} - \hat{r}_{ai})^2}{2\sigma^2}. \end{aligned} \tag{12}$$

3.3 Review Generation Module of DualPC

In this subsection, we design a model for the review generation task given the preference data. In other words, we model the generated review as $\hat{C}_{ai} = g(a, i, r_{ai}; \varphi)$ with parameter set φ (Eq.2). The structure of the review generation is shown in Fig.3. It can be treated as an encoder-decoder structure.

Encoder Part. Each input record is associated with a user ID, an item ID, and the preference of r_{ai}. The encoder part includes an embedding layer to transform all related attributes into vectors, followed by a fusion layer to combine all the vectors. Specifically, the embedding layer embeds the user and item related attributes, including user embeddings, item embeddings, as well as the sentiment reflected in the preference. Let $\mathbf{P} \in \mathbb{R}^{D\times|\mathcal{U}|}$ and $\mathbf{Q} \in \mathbb{R}^{D\times|\mathcal{V}|}$ denote the parameters of the free user and item embedding matrices in the review generation task. Then, user a's embedding vector is the a^{th} column of $\mathbf{P}$, denoted as $\mathbf{p}_a$. Similarly, item i's embedding vector is $\mathbf{q}_i$, i.e., the i^{th} column of $\mathbf{Q}$.

Each preference presents in discrete values, by treating each preference r_{ai} as this user's sentiment towards items, we use one-hot encoding to convert the preferences into vectors. We use $\mathbf{e}(r_{ai}) \in \{0, 1\}^K$ as the representation of the vectorized preference, with the vector length K equals the number of discrete values. E.g., in an explicit preference system with user preferences range from 1 to 5, K equals 5. And in implicit feedback based systems with user browsing behavior, add to cart behavior and the buying behavior, K=3. Each element of $\mathbf{e}(r_{ai}) \in \{0, 1\}^K$ denotes whether this user performs the corresponding action to the item. After that, we fuse all the embeddings as the output of the encoder as:

$$\mathbf{a} = [\mathbf{e}(r_{ai}), \mathbf{p}_a \circ \mathbf{q}_i]. \tag{13}$$

Decoder Part. The decoder part is built upon LSTM units, since it could better handle long sequences. Specifically, an LSTM uses a vector to represent information for the current step, and recursively updates the next hidden vector based on the previous hidden vector as well as the current input. Let $\mathbf{k}_{ai}^t$ denote the k^{th} hidden state of the LSTM structure, with the initial hidden state $\mathbf{k}_{ai}^0$=$\mathbf{a}$ is the output of the encoder part in Eq.13. At each time t, the LSTM takes the previous hidden state $\mathbf{k}_{ai}^{t-1}$, the t^{th} word representation of the review C_{ai} as input, and updates the current state $\mathbf{k}_{ai}^t$ as:

$$\mathbf{k}_{ai}^t = LSTM(\mathbf{k}_{ai}^{t-1}, \mathbf{e}_{ai}^t; \varphi_{LSTM}), \tag{14}$$

where φ_{LSTM} is the trainable parameters set of the LSTM structure, $\mathbf{e}_{ai}^t$ is the word embedding of the t^{th} word in C_{ai}. $\mathbf{e}_{ai}^t$ can be obtained by using word C_{ai}^t to retrieve the word embedding matrix $\mathbf{E}$. The word embedding matrix $\mathbf{E}$ is the same as the pre-trained word embeddings in the preference prediction part.

To get the conditional probability of each word at time t, we first map the output of the LSTM to the vocabulary space. Then, we use a softmax function to model the distribution of the output word C_{ai}^t=w as:

$$p(C_{ai}^t|C_{ai}^{<t}, r_{ai}; \varphi) = softmax_{C_{ai}^t}(\mathbf{F} \cdot \mathbf{k}_{ai}^t), \tag{15}$$

where $\mathbf{F} \in \mathbb{R}^{(|V| \times d)}$ is the trainable parameters, and with $|V|$ is the size of the vocabulary, and d=$|\mathbf{k}|_{ai}^t$ is the dimension of the hidden state of the LSTM. The ground truth word C_{ai}^t corresponds to the w^{th} word in the vocabulary and the conditional probability of C_{ai}^t is $p(C_{ai}^t|C_{ai}^{<t}, r_{ai}; \varphi)$.

As the conditional probability of review C_{ai} can be calculated with $\prod_{t=1}^{|C_{ai}|} p(C_{ai}^t|C_{ai}^{<t}, r_{ai}; \varphi)$, maximizing the likelihood of all the training reviews is equivalent to minimizing the Negative Likelihood Loss (NLL) equation (Eq.2) as:

$$l_2 = -\sum_{x \in \mathcal{X}} p(C_{ai}|r_{ai}; \varphi) = -\sum_{x \in \mathcal{X}} \sum_{t=1}^{|C_{ai}|} log(p(C_{ai}^t|C_{ai}^{<t}, r_{ai}; \varphi). \tag{16}$$

3.4 Model Training

In this section, we introduce how to train our proposed dual learning based DualPC framework of two modules: preference prediction and review generation. However, according to Eq.5 and Eq.6 of the two optimization functions, both functions have the duality loss (Eq.4). As the conditional probabilities of $p(r_{ai}|C_{ai})$ and $p(C_{ai}|r_{ai})$ have been modeled in the preceding two subsections, we first focus on the marginal preference distribution and review distribution. Then, we give a detailed training algorithm.

Marginal Distribution. Without the review data as input, we use a classical collaborative filtering model: Probabilistic Matrix Factorization (PMF) [21] to model the marginal distribution. Specifically, given users' historical preference records $\mathcal{R}$, we train a PMF model first, then we adopt the learned PMF model to calculate the empirical marginal distribution of the real preferences. Let $PMF(a, i)$ denotes the predicted preference of a user-item pair (a, i) under the PMF model with training data $\mathcal{R}$, the marginal distribution of the preference data is:

$$\begin{aligned} p(\mathcal{R}) &= \prod_{x \in \mathcal{X}} \mathcal{N}(r_{ai}|PMF(a, i), \sigma^2) \\ &= \prod_{x \in \mathcal{X}} \frac{1}{\sqrt{2\pi\sigma^2}} exp(\frac{-(r_{ai} - PMF(a, i))^2}{2\sigma^2}), \end{aligned} \tag{17}$$

where σ^2 is a hyperparameter that needs to be tuned.

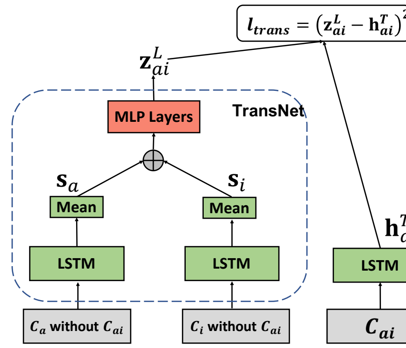


Figure 4: The structure of TransNet for review representation for review representation at test stage.

For the training data of reviews, we resort to LSTM to approximate the marginal distribution of reviews. The marginal distribution of the review distribution is modeled as

$$p(C) = \prod_{x \in \mathcal{X}} \prod_{t=1}^{|C_{ai}|} p(C_{ai}^t|C_{ai}^{<t}), \tag{18}$$

where C_{ai}^t is the t^{th} word in C_{ai}, $|C_{ai}|$ denotes the number of words in C_{ai}, and the index $< i$ indicates $1, 2, ..., t-1$.

After modeling the marginal distribution, the duality loss in Eq.4 can be computed in detail as:

$$\begin{aligned} l_{duality} = \sum_{x \in \mathcal{X}} [logp(r_{ai}) + \sum_{t=1}^{|C_{ai}|} logp(C_{ai}^t|C_{ai}^{<t}, r_{ai}; \varphi) \\ - \sum_{t=1}^{|C_{ai}|} logp(C_{ai}^t|C_{ai}^{<t}) - log(p(r_{ai}|C_{ai}; \theta))]^2. \end{aligned} \tag{19}$$

Please note that, in the duality loss, both the marginal distributions of preferences and reviews are computed in advance, and are not involved in the model training process of the DualPC framework. In fact, the computation of the marginal distributions is modeled to regularize the duality based loss, such that the conditional probabilities are better learned in the modeling process.

With the pre-computed marginal distributions, we combine Eq.12 and Eq.19 to obtain the optimization functions of preference prediction as follows:

$$\begin{aligned} &\min_{\theta} l_P = l_1 + \lambda_1 l_{duality} \\ &= \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} [log(\sqrt{2\pi\sigma^2}) + \frac{(r_{ai} - \hat{r}_{ai})^2}{2\sigma^2} + \lambda_1 l_{duality}], \end{aligned} \tag{20}$$

where $\theta = [\mathbf{U}, \mathbf{V}, [\mathbf{W}^l, \mathbf{b}^l]_{l=1}^L, \mathbf{w}', b, \theta_{LSTM}]$.

Similarly, the optimization function of review generation is obtained by combining the conditional review generation loss Eq.16, and the duality loss in Eq.19:

$$\begin{aligned} &\min_{\varphi} l_C = l_2 + \lambda_2 l_{duality} \\ &= \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} [-\sum_{t=1}^{|C_{ai}|} log(p(C_{ai}^t|C_{ai}^{<t}, r_{ai}; \varphi) + \lambda_2 l_{duality}], \end{aligned} \tag{21}$$

where $\varphi = [\mathbf{P}, \mathbf{Q}, \mathbf{F}, \varphi_{LSTM}]$. We show the detailed algorithm of the DualPC in Algorithm 1.

3.5 How to Test the DualPC Framework

In the training stage, we model the preference function $\hat{r}_{ai}$=$f(a, i, C_{ai}; \theta)$ with the associated review, and the review function $\hat{C}_{ai}$=$g(a, i, r_{ai}; \varphi)$,

Algorithm 1: Model Training of DualPC Framework.

Require:
The training dataset $\mathcal{X} = [\mathcal{U}, \mathcal{V}, \mathcal{R}, C]$;
Ensure:
Initialize all trainable parameters of preference prediction θ and review generation φ;
Precompute the marginal preference distribution with PMF and the marginal review distribution with LSTM;;
repeat
    Get a mini-batch of m user-item pairs, and corresponding preferences and reviews;
    for Each pair of input (a, i, C_{ai}, r_{ai}) in the mini-batch **do**
        Compute the conditional loss l_1 (Eq.12) and l_2 (Eq.16);
        Compute the duality loss $l_{duality}$(Eq.19);
        Compute the objective function of preference prediction in Eq.20 and review generation in Eq.21;
    end for
    Optimize θ to minimize preference prediction loss;
    Optimize φ to minimize review generation loss;
until Convergence of parameters.

and learn the parameter set θ and φ. In the test stage, for each candidate user-item pair (a, i), we do not have either the preference information or the review data. In this subsection, we would show how to predict the preferences and generate reviews during the testing process. Specifically, we would first borrow the key ideas of TransNet, a review based preference prediction model [3]. TransNet learns to transform the historical reviews of users and items, into a pair-wise user-item review representation at the test time when the user-item review is not available. Based on TransNet, we would first predict users' preferences at the test stage with the pair-wise user-item review representation. Then, the predicted preferences would be approximated as users' preferences and feed into the review generation function for review generation at the test time. In the following, we would give a brief introduction of TransNet, and show how the key ideas of TransNet can be used in our DualPC framework for the model test.

Let C_a denotes all the historical reviews of user a in the training data, and C_i denotes all the historical reviews of item i. For each user-item pair (a, i), as the real user-item review C_{ai} is the most important for predicting the preference $\hat{r}_{ai}$, TransNet is proposed to approximate the real user-item review representation with a transformation structure [3]. For each user-item review C_{ai}, it is first sent to LSTMs to get the review semantic representation $\mathbf{h}^T_{ai}$ as shown in Eq.7. Therefore, for each user-item pair (a, i), TransNet needs to learn a function $\hat{z}_{ai}=s(a, i, C_a, C_i)$ to approximate the real review representation $\mathbf{h}^T_{ai}$. We show the overall structure of TransNet in Fig. 4. Specifically, to mimic $\mathbf{h}^T_{ai}$, TransNet first takes user i's reviews and item i's reviews in the training data and get the review semantic representation $\mathbf{s}_a$ and $\mathbf{s}_i$ as:

$$\begin{aligned}\mathbf{s}_a &= \frac{1}{|C_a| - 1} \sum_{k:[r_{ak}=1, k \neq i]} LSTM(C_{ak}),\\ \mathbf{s}_i &= \frac{1}{|C_i| - 1} \sum_{b:[r_{bi}=1, b \neq a]} LSTM(C_{bi}),\end{aligned} \tag{22}$$

where the LSTM is the same as the one in preference prediction part so that there are no additional parameters. We have to note that, the current review C_{ai} is omitted for both user and item review representation to avoid the leakage of the optimization goal during testing. Next, we concatenate $\mathbf{s}_a$ and $\mathbf{s}_i$ and send the result to MLPs for the representation $\mathbf{z}^L_{ai}$ as :

$$\begin{aligned}\mathbf{z}^0_{ai} &= [\mathbf{s}_a, \mathbf{s}_i]\\ \mathbf{z}^l_{ai} &= ReLU(\mathbf{W}^l_M \mathbf{z}^{l-1}_{ai}),\ l = 1, .., L.\end{aligned} \tag{23}$$

The output of MLPs, i.e., $\mathbf{z}^L_{ai}$, is designed to approximate the true review representation $\mathbf{h}^T_{ai}$. Therefore, the optimization function is defined as:

$$l_{trans} = (\mathbf{z}^L_{ai} - \mathbf{h}^T_{ai})^2. \tag{24}$$

And in TransNet, the trainable parameters ϕ=$[\mathbf{W}^l_M]^L_{l=1}$.

Algorithm 2: TransNet part for model prediction.

Require:
The training dataset $\mathcal{X}$=$[\mathcal{U}, \mathcal{V}, \mathcal{R}, C]$;
Parameter set of θ with θ_{LSTM}.
Ensure:
Initialize the parameter set ϕ of TransNet.
repeat
    Get a mini-batch of m user-item pairs, and corresponding reviews;
    for Each pair of input (a, i, C_{ai}) in the mini-batch **do**
        Compute the real review representation $\mathbf{h}^T_{ai}$;
        Compute the approximated review representation $\mathbf{z}^L_{ai}$ (Eq.23);
        Update the TransNet loss in Eq.24
    end for
    Optimize the parameter set of ϕ;
until Convergence of model parameters.

Model Prediction. In the test stage, for each candidate user-item pair (a, i), we first use the TransNet to approximate the unobserved review representation $\mathbf{h}^T_{ai}$ as $\mathbf{z}^L_{ai}$. Then, the predicted preference $\hat{r}_{ai}$ is inferred as:

$$\hat{r}_{ai} = f(a, i, C_{ai}; \theta) = f(a, i, \mathbf{h}^T_{ai}; \theta) \approx f(a, i, \mathbf{z}^L_{ai}; \theta). \tag{25}$$

With the predicted preference $\hat{r}_{ai}$, we first round this value to the nearest integer $round(\hat{r}_{ai})$. Then, we input the triplet $(a, i, round(\hat{r}_{ai}))$ to the review generation function to get the generated review as:

$$\hat{C}_{ai} = g(a, i, r_{ai}; \varphi) \approx g(a, i, round(\hat{r}_{ai}); \varphi). \tag{26}$$

4 EXPERIMENTAL SETUP

Datasets. We evaluate our proposed method on two datasets: *Amazon Books*[20] and *Amazon Electronics*[31] from the Amazon dataset[1]. The preferences range from 1 to 5. To reduce the impact of complex symbols on language modeling, we keep the reviews that only contain symbols of the comma and period. After that, we require users and items have more than one review, and the frequency of words should be more than 3. After data preprocessing, we select 85% of the data for training, 5% for validation and the remaining 10% as the test data. For each review, we add the 'SOS' and 'EOS' flags at the start and end position, respectively. To keep the reviews have the same length as the longest reviews, we add the 'PAD' flags to the reviews with shorter lengths. The statistics of the datasets are shown in Table 1.

[1] http://jmcauley.ucsd.edu/data/amazon/

Table 1: The statistics of the two datasets.

Dataset	*Books*	*Electronics*
Users	185703	140766
Items	126137	72984
Training preferences	578341	288698
Test preference	58801	25446
preference Density	0.0025%	0.0028%
Words	33806	17132
Reviews Maximum Length	33	32

Implementation Details. We implement DualPC by using PyTorch[2]. The models are trained on NVIDIA GTX1080Ti. The dimension of the word embedding $\mathbf{E}$ is set as 150, and it is pre-trained with the Gensim word2vec[3]. The hidden state size of both LSTM modules in the preference prediction and review generation is set as 128 for the Amazon Books and 256 for Amazon Electronics (Eq.7 and Eq.14). The number of the layers of the MLPs is set as 3 in the preference prediction (Eq.9) process. Specifically, the hidden state size of each layer of the MLPs is half of the previous one. In the test stage, the number of the layers of MLPs is set as 4. In the first 3 layers, the hidden state size of each layer of the MLPs is half of the dimension of the previous one. And the last layer of the hidden dimension in MLP (i.e., $\mathbf{z}_{ai}^L$)is set as the same size of the hidden state of LSTM (i.e., $\mathbf{h}_{ai}^T$). To initialize the DualPC model, we randomly inatialize all parameters such as $[\mathbf{U}, \mathbf{V}, \mathbf{P}, \mathbf{Q}, [\mathbf{W}^l, \mathbf{b}^l]_{l=1}^3, \mathbf{w}', b, \mathbf{F}, [\mathbf{W}_M^l]_{l=1}^4]$ following a normal distribution with mean as 0 and variance as 0.01. We set [λ_1=0.2, λ_2=0.4] and [λ_1=0.2, λ_2=0.2] for Amazon Books and Amazon Electronics, respectively. We use Adam optimizer with a learning rate 0.001, and the regularization coefficient is set as 0.001. Last but not least, for the variance σ^2 hyperparameter of Gaussian distribution in Eq.11 and Eq.17, we set the variance parameter as the squared error of the test preferences learned from PMF.

5 PREFERENCE PREDICTION TASK

Baselines and Evaluations. The baselines can be divided into two groups:

Preferences Only. The methods in this group only consider the preference data. Like *PMF*[21] and *NeuMF*[10]. PMF is based on the matrix factorization for user preference prediction, which projects both user and item into a low latent space. NeuMF advances PMF by modeling the non-linear interaction between the users and items.

Preferences and Reviews. This methods in this group not only consider the preference data, but also the reviews data. Like *FM*[23], *DeepCoNN*[45], *NARRE*[4], *TransNet*[3] and *MRG*[30]. FM is a feature enriched CF model by modeling both user and item interaction behavior, as well as the user and item features. In review enhanced recommendation, each user's (item's) feature is represented as a vector of words. For DeepCoNN and NARRE, the preference is learned by modeling the non-linear interactions between user and item semantic vector. The difference between them is how to learn the user and item semantic vector. DeepCoNN uses two parallel CNN based networks for user and item review semantic embedding. Instead of modeling each user by equally combining the representations of each user's historical reviews, NARRE advances DeepCoNN by attentively selecting important reviews for recommendation. As the target user-item review is not available at test time, TransNet advances DeepCoNN by introducing an additional latent layer representing the target user-item pair. MRG is a state-of-the-art multi-task learning model for both preference prediction and review generation. The multimodal data are used to better inform the review generation process. In practice, as we do not have the multimodal information, we simplify this model with no additional multimodal input.

We select the Root Mean Square Error(RMSE) as the evaluation metrics, wich is calculated as:

$$RMSE = (\frac{1}{|\mathcal{D}|}\sum_{\mathcal{D}}(r_{ai} - \hat{r}_{ai})^2)^{\frac{1}{2}}, \tag{27}$$

where r_{ai} is the real preference and $\hat{r}_{ai}$ is the predicted preference. $\mathcal{D}$ is the test set for evaluation.

Overall Performance Comparison. Table 2 shows the RMSE results of all models on two datasets with varying dimension size D. We can observe that our proposed DualPC framework outperforms all baselines under any dimension D, showing the effectiveness of correlating these two tasks by duality learning. Among all baselines, DeepCoNN and NARRE utilize the reviews as auxiliary information to alleviate the data sparsity problem. Therefore, these two review enhanced models show better preference prediction accuracy compared to CF baselines of PMF and NeuMF. As the target user-item review contributes most in recommendation, TransNet advances DeepCoNN by introducing an additional latent layer to approximate the representation of the target user-item review. Among all the baselines, we observer that though MRG jointly models the two tasks with a multi-task setting, it does not perform better than most baselines. We speculator that in MRG model with user preference learning, the predicted preference is only modeled with the user free emebedding and item free embedding without any content enriched modeling techniques. Therefore, MRG has inferior performance on preference prediction task. As D increases, the performance of our DualPC model also increases, and DualPC always show the best performance. For example, when D=128, DualPC increases the best baseline about 1.5% on RSME metric, showing the effectiveness of enhancing preference prediction with duality based loss. Based on the above analysis, we set D=128 in the following experiments.

6 REVIEW GENERATION TASK

Baselines and Evaluations. We compare our proposed model with the following baselines, *Naive LSTM*[29], *SRGM-s*[40], *Att2Seq*[46], *MRG*[30]. All the methods are mainly based on the encoder-decoder architecture. Naive LSTM reconstructs the input sentence with an LSTM structure, which is a generalized model and does not take any personalized information into consideration. SRGM-s advances the Naive LSTM by modeling the preferences in the review generation process to guide the generated reviews to be consistent with preferences. Att2Seq encodes the user latent embedding, item latent embedding, and the preferences. And MRG is a multi-task learning model for both user preference prediction and review generation.

Since it is a language generation problem, we select BLEU[22] and ROUGE[15] to evaluate the model performance. They both are calculated based on the overlapping content of the candidate generated reviews and the real reviews with different methods.

[2]https://pytorch.org
[3]https://radimrehurek.com/gensim/models/word2vec.html

Table 2: Preference prediction task: RMSE comparisons of different dimension size D on two datasets.

Models	Amazon Books				Amazon Electronics			
	D=16	D=32	D=64	D=128	D=16	D=32	D=64	D=128
PMF	0.9480	0.9480	0.9470	0.9479	1.0994	1.0996	1.0995	1.0994
NeuMF	0.9354	0.9147	0.8958	0.8665	1.0657	1.0525	1.0297	1.0267
FM	0.9302	0.9161	0.8945	0.8722	1.0429	1.0348	1.0268	1.0155
DeepCoNN	0.9290	0.9132	0.9002	0.8621	1.0333	1.0284	1.0092	0.9903
NARRE	0.9246	0.9082	0.8931	0.8507	1.0281	1.0102	1.0041	0.9819
TransNet	0.9273	0.9107	0.8974	0.8624	1.0327	1.0229	0.9947	0.9828
MRG	0.9380	0.9238	0.9047	0.8751	1.0984	1.0848	1.0613	1.0582
DualPC	**0.9093**	**0.8914**	**0.8788**	**0.8376**	**0.9825**	**0.9743**	**0.9672**	**0.9676**

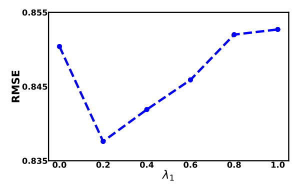
However, the BLEU score is precision-oriented, and ROUGE scores contain the Precision, Recall, and F-measure parts. Moreover, the ROUGE has many different strategies to measure the overlapping content between two sentences. In this experiment, we will report the results based on Recall, Precision, and F-measure of ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-SU4 and the BLEU scores.

Overall Performance Comparison. Table 3 and Table 4 show the overall performance of all models on review generation task. Based on the results, we have following observations.

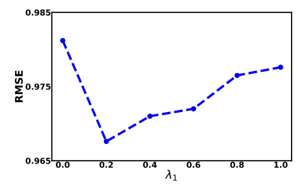
First of all, DualPC outperforms all comparison methods on the two datasets, and achieves best performance on BLEU and F-measure of ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-SU4 metrics. Among all baselines, we can observe that Naive LSTM and SRGM-s achieve better performance on ROUGE Recall. We speculate that the long sentence can be generated more easily with less external information. However, our proposed DualPC achieves best performance on ROUGE F-measure, indicating that DualPC is capable of generating more suitable reviews. We believe that the possible reasons are tow-fold. On one hand, DualPC encodes the interaction of user and item, and the preference information, providing sufficient information for the decoder to generate personalized reviews that are correlated with items. On the other hand, the duality constrains of the two tasks make the user and item embedding learning more precise. Therefore, DualPC also outperforms Att2Seq that also takes similar input, but with the user and item embedding learning separately from the preference history.

Among all baselines, MRG is the current state-of-the-art method. It leverages the multi-task learning method and share the user and item embeddings between two tasks, so that the relationship between preference prediction and review generation can be better verified. Different from MRG, our proposed DualPC takes the dual learning method into consideration, so that we can not only better analyze the interaction between preference prediction and review generation, but also enhance the model performance on one of the task with the results on the other task. Thus, we can observe that our proposed DualPC achieves the best performance compared with other baselines.

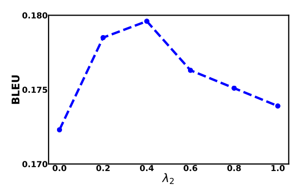
Case Study. Table 5 shows some examples of reviews generated with the baseline models based on Amazon Electronics. We don't show the results produced by Naive LSTM and SRGM-s since they don't take any personalized information into account. The real cases are from three pairs of user-item, with the real preferences and reviews are also shown for better understanding. From the results, we can observe that the reviews generated by DualPC are more fluent with fewer grammar errors. The bold words in the sentences are the keywords that are related to the real reviews and the phrases that persuade customers to buy this product. Comparing with the baseline models, by using the probabilistic correlation duality constraints, the reviews generated by our DualPC framework looks closer to the real reviews and are more persuasive.



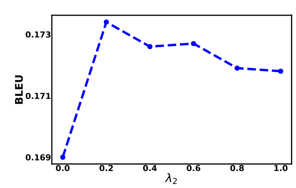
(a) RMSE w.r.t λ_1 on Amazon Books



(b) RMSE w.r.t λ_1 on Amazon Electronics



(c) BLEU w.r.t λ_2 on Amazon Books



(d) BLEU w.r.t λ_2 on Amazon Electronics

Figure 5: The influence of the λ_1 and λ_2 on the two datasets.

Sensitivity of Parameters. In DualPC, according to the optimization functions of Eq.20 and Eq.21 for the two tasks, λ_1 and λ_2 are two important parameters of the duality based loss terms that need to be tuned. The larger values of the two parameters, the more the probabilistic duality terms play roles in the optimization process. Specifically, when λ_1=λ_2=0, DualPC degenerates to classical preference prediction models and review generation models without any duality constraints.

To investigate the effects of λ_1 and λ_2, we first tune the λ_1 among [0, 0.2, 0.4, 0.6, 0.8, 1.0] with λ_2=0.2. Fig.5 (a) and (b) show the performance of the preference prediction task with RMSE metric as λ_1 varies. The smaller value of RMSE, the better the preference prediction performance. For both datasets, as λ_1 increases from 0 to 0.2, the preference prediction performance increases. As we further increases λ_1, the performance drops. Therefore, we set λ_1=0.2 for both datasets. Fig.5 (c) and (d) show the performance of the review generation task as the λ_2 varies (λ_1=0.2). It is evaluated based on the BLUE metric, with the larger value means the better performance. We observe a similar phenomenon that the performance increases first and then drops with the increase of λ_2. When setting λ_1=0.2 and λ_2=0.4, the DualPC performs best on Amazon Books an Amazon Electronics, respectively. Therefore, we choose [λ_1=0.2, λ_2=0.4] and

Table 3: Review generation task: ROUGE and BLEU comparisons on Amazon Books dataset.

Models	ROUGE-1			ROUGE-2			ROUGE-L			ROUGE-SU4			BLEU
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	
Naive LSTM	0.1455	0.1801	0.1810	0.0227	**0.0249**	0.0224	0.1455	0.1398	0.1438	0.0523	**0.0587**	0.0512	0.1645
SRGM-s	0.1774	0.1866	0.1744	0.0229	0.0240	0.0222	0.1441	**0.1508**	0.1409	0.0518	0.0542	0.0493	0.1677
Att2Seq	0.1843	**0.1875**	0.1784	0.0241	0.0242	0.0229	0.1488	0.1505	0.1433	0.0544	0.0545	0.0508	0.1722
MRG	0.1898	0.1834	0.1793	0.0245	0.0239	0.0234	0.1496	0.1478	0.1435	0.0554	0.0537	0.0511	0.1757
DualPC	**0.1910**	0.1842	**0.1809**	**0.0255**	0.0240	**0.0235**	**0.1537**	0.1469	**0.1446**	**0.0568**	0.0532	**0.0517**	**0.1796**

Table 4: Review generation task: ROUGE and BLEU comparisons on Amazon Electronics dataset.

Models	ROUGE-1			ROUGE-2			ROUGE-L			ROUGE-SU4			BLEU
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	
Naive LSTM	0.1595	**0.1839**	0.1627	0.0161	**0.0186**	0.0163	0.1288	**0.1480**	0.1289	0.0447	**0.0530**	0.0446	0.1575
SRGM-s	0.1611	0.1724	0.1569	0.0163	0.0175	0.0160	0.1289	0.1376	0.1271	0.0450	0.0484	0.0435	0.1675
Att2Seq	0.1639	0.1714	0.1604	0.0167	0.0173	0.0161	0.1319	0.1374	0.1284	0.0464	0.0484	0.0441	0.1651
MRG	**0.1700**	0.1706	0.1621	0.0168	0.0167	0.0162	0.1328	0.1360	0.1283	0.0469	0.0475	0.0441	0.1712
DualPC	0.1684	0.1716	**0.1635**	**0.0170**	0.0171	**0.0163**	**0.1346**	0.1362	**0.1299**	**0.0478**	0.0481	**0.0449**	**0.1734**

Table 5: Case study of generated reviews based on Amazon Electronics.

Case 1	*preference*: 1; *Review*: I read the description for this product which said it would be compatible with my macbook and upon receiving it. It did not work at all.
Att2Seq	This case was easy for my ipad air to the tablet and the keyboard was just right on it.
MRG	This case works just fine with the iphone or ipad mini.
DualPC	This is **not an official apple** product that was **not compatible** with my **tablet**.
Case 2	*preference*: 5; *Review*: The case is sturdy the keyboard is perfect for my kids to do their studies with and type on the ivery easy to use.
Att2Seq	Have been using this for over year and all the buttons are comfortable and accurate.
MRG	This is great product. It is very easy to use and very comfortable to use.
DualPC	This is **great product**. It is very **durable** and **protects the ipod perfectly**. The **quality is great** and it is very **versitile**.
Case 3	*preference*: 5; *Review*: Great picture love the smart tv features. This is great tv for the money. HD picture quality is just as good as have seen around.
Att2Seq	Great price and easy to use. Use it for my garmin gps and this can be me for the money.
MRG	This is my second cheetah mount. This is my second one and it works well. I have used it to replace the one that came with the tv.
DualPC	The picture is very **clear** and the picture is **not the best**. But I am not sure about it. But it has **worked great** with my **sony bravia tv**.

[λ_1=0.2, λ_2=0.2] as the default parameters on Amazon Books and Amazon Electronics, respectively.

We would like to discuss more about different experimental settings. When both λ_1 and λ_2 are set to zero, the duality constraints in the two prediction tasks are missing. Therefore, the two tasks are trained independently. E.g., no matter what value of λ_2, when λ_1=0, preference prediction results would be merely influenced by l_1 loss in Eq.20. Therefore, the preference prediction results are the same as the leftmost in Fig.5 (a) and (b) with λ_1=0 and is worse than DualPC with λ_1=0.2.

When λ_1 is non-zero, the duality constraints for preference prediction remain, while the review generation task doesn't have a duality regularization term. Then, the performance of review generation is equal to the leftmost part, as shown in Fig.5 (c) and (d) with λ_2=0. And the preference prediction tasks would perform worse compared to non-zero λ_1 results in Fig.5 (a) and (b). When λ_2 is non-zero, the analysis is similar to the previous analysis.

7 CONCLUSIONS

In this paper, we argued that in review based recommender systems, the user preference prediction task and the review generation task are presented in dual forms. To this end, we proposed a dual learning based framework of DualPC, to model the probabilistic correlation among these two tasks. Specifically, the duality correlation between two tasks is turned to a duality based regularization of each task. We further designed a structure of how to test our model when both the user-item preference information and the review data are not available at the test stage. We conducted extensive experimental results on two real-world datasets to show the effectiveness of our proposed model for recommendation.

ACKNOWLEDGMENTS

This work was supported in part by grants from the National Key Research and Development Program of China(Grant No.2017YFB0803301), the National Natural Science Foundation of China(Grant No.61725203, 61972125, U19A2079, 61722204, 61932009 and 61732008).

REFERENCES

[1] Amjad Almahairi, Kyle Kastner, Kyunghyun Cho, and Aaron Courville. 2015. Learning distributed representations from reviews for collaborative filtering. In *RecSys*. 147–154.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

[3] Rose Catherine and William Cohen. 2017. Transnets: Learning to transform for recommendation. In *RecSys*. 288–296.

[4] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural attentional rating regression with review-level explanations. In *WWW*. 1583–1592.

[5] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2019. Social Attentional Memory Network: Modeling Aspect- and Friend-Level Differences in Recommendation. In *WSDM*. 177–185.

[6] Xu Chen, Hanxiong Chen, Hongteng Xu, Yongfeng Zhang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2019. Personalized Fashion Recommendation with Visual Explanations based on Multimodal Attention Network: Towards Visually Explainable Recommendation. In *SIGIR*. 765–774.

[7] Zhongxia Chen, Xiting Wang, Xing Xie, Tong Wu, Guoqin Bu, Yining Wang, and Enhong Chen. 2019. Co-Attentive Multi-Task Learning for Explainable Recommendation. In *IJCAI*. 1237–2143.

[8] Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In *NIPS*. 820–828.

[9] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. Trirank: Review-aware explainable recommendation by modeling aspects. In *CIKM*. 1661–1670.

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

[11] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *SIGKDD*. 426–434.

[12] Wu Le, Chen Lei, Hong Richang, Fu Yanjie, Xie Xing, and Wang Meng. 2019. A Hierarchical Attention Model for Social Contextual Image Recommendation. *TKDE* (2019).

[13] Piji Li, Zihao Wang, Lidong Bing, and Wai Lam. 2019. Persona-Aware Tips Generation. In *WWW*. 1006–1016.

[14] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural rating regression with abstractive tips generation for recommendation. In *SIGIR*. 345–354.

[15] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *ACL*. 74–81.

[16] Yujie Lin, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten de Rijke. 2019. Explainable fashion recommendation with joint outfit matching and comment generation. *TKDE* (2019).

[17] Guang Ling, Michael R Lyu, and Irwin King. 2014. Ratings meet reviews, a combined approach to recommend. In *RecSys*. 105–112.

[18] Yichao Lu, Ruihai Dong, and Barry Smyth. 2018. Coevolutionary recommendation model: Mutual learning between ratings and reviews. In *WWW*. 773–782.

[19] Yichao Lu, Ruihai Dong, and Barry Smyth. 2018. Why I like it: multi-task learning for recommendation and explanation. In *RecSys*. 4–12.

[20] Julian McAuley and Alex Yang. 2016. Addressing complex and subjective product-related queries with customer reviews. In *WWW*. 625–635.

[21] Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic matrix factorization. In *NIPS*. 1257–1264.

[22] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*. 311–318.

[23] Steffen Rendle. 2010. Factorization machines. In *ICDM*. 995–1000.

[24] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*. 452–461.

[25] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *RecSys*. 297–305.

[26] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Representation learning of users and items for review rating prediction using attention-based convolutional neural network. In *SDM Workshop*.

[27] Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*. 3776–3784.

[28] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. LSTM Neural Networks for Language Modeling. In *INTERSPEECH*. 194–197.

[29] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*. 3104–3112.

[30] Quoc-Thuan Truong and Hady W.Lauw. 2019. Multimodal Review Generation for Recommender Systems. In *WWW*. 1864–1874.

[31] Mengting Wan and Julian McAuley. 2016. Modeling ambiguity, subjectivity, and diverging viewpoints in opinion question answering systems. In *ICDM*. 489–498.

[32] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge-aware Graph Neural Networks with Label Smoothness Regularization for Recommender Systems. In *KDD*. 968–977.

[33] Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019. Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation. In *WWW*. 2000–2010.

[34] Xiting Wang, Yiru Chen, Jie Yang, Le Wu, Zhengtao Wu, and Xing Xie. 2018. A Reinforcement Learning Framework for Explainable Recommendation. In *ICDM*. 587–596.

[35] Le Wu, Yong Ge, Qi Liu, Enhong Chen, Richang Hong, Junping Du, and Meng Wang. 2017. Modeling the evolution of users' preferences and social links in social networking services. *TKDE* 29, 6 (2017), 1240–1253.

[36] Le Wu, Yong Ge, Qi Liu, Enhong Chen, Bai Long, and Zhenya Huang. 2016. Modeling users' preferences and social links in social networking services: a joint-evolving perspective. In *AAAI*. 279–286.

[37] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019. A Neural Influence Diffusion Model for Social Recommendation. In *SIGIR*. 235–244.

[38] Yingce Xia, Tao Qin, Wei Chen, Jiang Bian, Nenghai Yu, and Tie-Yan Liu. 2017. Dual supervised learning. In *ICML*. 3789–3798.

[39] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*. 2048–2057.

[40] Hongyu Zang and Xiaojun Wan. 2017. Towards automatic generation of product reviews from aspect-sentiment scores. In *ICNLG*. 168–177.

[41] Kun Zhang, Guangyi Lv, Le Wu, Enhong Chen, Qi Liu, Han Wu, Xing Xie, and Fangzhao Wu. 2019. Multilevel Image-Enhanced Sentence Representation Net for Natural Language Inference. *Trans. SMC: System* (2019).

[42] Tao Zhang, Jin Zhang, Chengfu Huo, and Ren Weijun. 2019. Automatic Generation of Pattern-controlled Product Description in E-commerce.. In *WWW*. 2355–2365.

[43] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *SIGIR*. 83–92.

[44] Lujun Zhao, Kaisong Song, Changlong Sun, Qi Zhang, Xuanjing Huang, and Xiaozhong Liu. 2019. Review Response Generation in E-Commerce Platforms with External Product Information.. In *WWW*. 2425–2435.

[45] Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *WSDM*. 425–434.

[46] Ming Zhou, Mirella Lapata, Furu Wei, Li Dong, Shaohan Huang, and Ke Xu. 2017. Learning to Generate Product Reviews from Attributes. In *EACL*. 623–632.

[47] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*. 2223–2232.