ReXPlug: Explainable Recommendation using Plug and Play Language Model

Deepesh V. Hada[*]
deepeshhada@iisc.ac.in
Indian Institute of Science
Bangalore, Karnataka, India

Vijaikumar M.[*]
vijaikumar@iisc.ac.in
Indian Institute of Science
Bangalore, Karnataka, India

Shirish K. Shevade
shirish@iisc.ac.in
Indian Institute of Science
Bangalore, Karnataka, India

ABSTRACT

Explainable Recommendations provide the reasons behind why an item is recommended to a user, which often leads to increased user satisfaction and persuasiveness. An intuitive way to explain recommendations is by generating a synthetic personalized natural language review for a user-item pair. Although there exist some approaches in the literature that explain recommendations by generating reviews, the quality of the reviews is questionable. Besides, these methods usually take considerable time to train the underlying language model responsible for generating the text.

In this work, we propose ReXPlug, an end-to-end framework with a plug and play way of explaining recommendations. ReXPlug predicts accurate ratings as well as exploits Plug and Play Language Model to generate high-quality reviews. We train a simple sentiment classifier for controlling a pre-trained language model for the generation, bypassing the language model's training from scratch again. Such a simple and neat model is much easier to implement and train, and hence, very efficient for generating reviews. We personalize the reviews by leveraging a special jointly-trained cross attention network. Our detailed experiments show that ReXPlug outperforms many recent models across various datasets on rating prediction by utilizing textual reviews as a regularizer. Quantitative analysis shows that the reviews generated by ReXPlug are semantically close to the ground truth reviews, while the qualitative analysis demonstrates the high quality of the generated reviews, both from empirical and analytical viewpoints. Our implementation[1] is available online.

CCS CONCEPTS

• **Information systems** → **Collaborative filtering**; **Personalization**; **Recommender systems**; • **Computing methodologies** → **Natural language generation**; **Transfer learning**.

KEYWORDS

Recommender Systems; Collaborative filtering; Transfer Learning

[*] Equal contribution. Listing order is random.
[1] https://github.com/deepeshhada/ReXPlug/

SIGIR '21, July 11–15, 2021, Virtual Event, Canada
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3462939

ACM Reference Format:
Deepesh V. Hada, Vijaikumar M., and Shirish K. Shevade. 2021. ReXPlug: Explainable Recommendation using Plug and Play Language Model. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3404835.3462939

1 INTRODUCTION

The overwhelming amount of data over the Internet demands efficient filtering of information. To address this overload problem, recommender systems perform product filtering across the web. The main aim of recommender systems is to predict whether a user will interact with an item or not, where these interactions can be of many forms. Collaborative filtering is one such effective and widely used technique which exploits the past user-item interactions to predict a rating for a user-item pair. The most common approach for collaborative filtering is to learn fixed-sized embedding vectors (latent features) to represent each user and item, and then predict unknown ratings using these embedding vectors. Matrix factorization is an early and popular model, which maps the users and items to their respective embeddings and applies an inner product on these latent features to capture the interaction. Deep learning models like NeuMF [14] also use such embeddings, but replace the inner product with a neural architecture that can learn an arbitrary function from data. These models predict ratings for a given pair, based on which, recommendations can be made to users.

Along with the available ratings, models can exploit another rich source of interaction information between users and items: user written reviews. A single review contains much more information about the concerned user-item pair than a single rating. Also, as they are abundantly available, a large amount of information can be extracted from them. Intuitively, reviews also explain *why* the user had bought the item, and what the user feels about the item after buying it, giving some potential explainability. DeepCoNN [46] is one of the first models to leverage neural networks for review-based collaborative filtering. Further modifications to DeepCoNN were in the form of TransNets [2], and attentive networks like MPCN [40] and NARRE [4].

Explainability in recommendations has been an essential aspect of personalized recommendation research. It addresses the problem of *why* by providing users with recommendation results and providing explanations behind the recommendations. In this way, it improves the transparency, persuasiveness, effectiveness, trustworthiness, and user satisfaction of the recommender systems. Extensive analysis has been done for providing reasons to users for the recommendations made. The explainability comes in many

Table 1: Comparison of generated reviews as explanations by ReXPlug with CAML [5]. In contrast to previous models, ReXPlug generates more *personalized* explanations. *(a)* and *(b)* demonstrate the specificity of explanations by ReXPlug. *(c)* shows the factual correctness of the explanation: *David Cage* is *Heavy Rain*'s writer and director.

	CAML	ReXPlug	Ground Truth Text
(a)	I love this *song*. It is a very nice song to listen to.	I love this ***song by Katy Perry*** she is one of my favorites and this song is amazing she did an awesome job on singing it. I can listen to her forever!!	It's ***Katy Perry's first hit***. Highly recommend it, a very different song but if you are collecting Katy Perry's music it is a must have!!!
(b)	This is a very nice *shirt* and the material is soft material is very nice.	This ***thermal*** will keep you warm. I recommend this thermal. Absolutely amazing value for money. I would purchase again if I had to.	This long sleeved ***thermal*** top fits great, looks good, and has held up well to wear and tear. With a little bleach it stays nice and white.
(c)	This is the first *game* that I have ever played. The graphics are amazing and the gameplay is not as good as the first one.	Will ***David Cage*** quit gaming business when he's offered a job as a Hollywood movie director? Only time will tell. But for now, the cinematics is a delight in their own right.	Same makers as ***Heavy Rain***. This game will bring you to tears. Worth the money. Stop reading reviews and buy it.

forms, as detailed in [44]. Most explainable recommendation models provide some pre-defined explanations like sentence templates [39, 42, 45] or association rules [11]. The most basic recommendation is when the predictions are explained by other "similar" items that the user is familiar with, which is the fundamental notion behind collaborative filtering. A simple text explanation, in this case, can be, *this product recommended to you is similar to the other products you liked before*. However, such simple explanations may not be persuasive enough for the user to buy the product.

In recent years, deep learning models have become successful in the domain of personalized recommendations [4, 5, 14, 40, 46]. However, the problem with these models is their black-box nature, which brings difficulty in the explanations. A more natural form of explanation is to generate synthetic text explanations using language models. In this variant, the fundamental aim is to generate sentences that resemble how the user has written reviews in the past. If the model somehow mimics how a user writes a review, the user could perhaps be more satisfied with the recommendation. Such an approach is challenging, mainly because of the noise in the text generation process, and is still in its early stage. There has been some recent work in trying to generate natural language explanations. The basic idea behind some of the older methods employing this form is to train sequence-to-sequence models based on user reviews from the training set and generate review-like sentences as explanations [5, 23]. These models' major drawback is training the language model, which generates text explanations from scratch, making training them computationally expensive and not cost-effective. Also, the generated explanations' fluency is questionable, as their respective language models are not trained on vast corpora.

With the advent of transformers [41], transfer learning approaches are the key in achieving such personalized explanations to recommendations, as pre-trained language models like BERT [9] and GPT-2 [36] can be fine-tuned to datasets of our choice. The proposed framework, ReXPlug, apart from predicting ratings for recommendations, builds on top of a huge pre-trained language model for controlled text generation for explanations. Transfer learning makes ReXPlug very efficient in terms of the training time and the number of trainable parameters while also providing high-quality and domain-specific explanations. Table 1 contrasts the explanations generated by ReXPlug and Co-Attentive Multi-Task Learning (CAML) [5] against the ground truth reviews. Unlike CAML, ReXPlug generates very specific reviews close to the ground truth review. The third generated review in Table 1 shows the factual correctness of explanations by ReXPlug.

This work makes the following major contributions:

1. We propose an end-to-end neural network framework – ReXPlug (EXplainable Recommendation using Plug and Play Language Model) for generating high-quality explainable recommendations by generating reviews on behalf of the user. ReXPlug is much more efficient in terms of the time required to train it and outperforms a state-of-the-art generative model on evaluation of explanations.

2. We empirically compare the Review Regularizer with Cross-Attention (RRCA), a sub-module of ReXPlug, with other state-of-the-art models for predicting ratings. RRCA outperforms these models on the task of rating prediction.

3. We perform ablation studies to demonstrate the effectiveness of the sub-models of ReXPlug.

4. We quantitatively analyze the generated reviews by
(i) automatic evaluation metrics like BLEU and Distinct scores;
(ii) computing Pearson Correlation Coefficient between the explanations and the ground truth reviews by ReXPlug; and
(iii) comparing the sentiments of the generated reviews with ground truth reviews.

2 RELATED WORK

2.1 Rating Prediction

For the task of rating prediction for recommendations, there exist a wide variety of approaches. These range from the traditional approach of Matrix Factorization [21], to neural-networks for collaborative filtering, to utilizing reviews for powerful recommendations. We use some of these models for comparisons.

Latent Factor models (LFMs) are a practical methodology for model-based collaborative filtering. Matrix Factorization [21] is a class of LFM which factorizes a user-item matrix to find latent space representations of users and items. Neural Matrix Factorization (NeuMF) [14] improves upon MF by modelling the user-item interaction with a neural network, instead of a simple inner product. Though simple, a major drawback of factorization based approaches is that they offer no explainability for the predicted recommendations. Moreover, these techniques do not leverage review information for better rating prediction.

Another class of recommendation models use textual reviews as inputs. Deep Co-operative Neural Network (DeepCoNN) [46] was one of the first deep learning methods proposed to learn item properties and user behaviour jointly from review text for predicting a user-item rating. It first concatenates all the reviews given by/to a user/item to form a review document, which represents users and items, by assuming that these reviews are independent of each other. To extract the latent features from the input user-item review documents, it uses TextCNN [20], an influential CNN-based architecture. TextCNN discovers latent features from the input review documents, followed by a neural network conditioned on these latent features to predict the rating. TransNets [2] unsurprisingly show that much of the predictive value of review text comes from reviews of the target user for the target item. In addition to using the user u and item i's review documents for extracting latent features, TransNets also uses the current review for regularization. It has two sub-networks, the first focuses on the given review sentiment, and the other being the same as DeepCoNN.

DeepCoNN makes a strong assumption of review independence. Neural Attentive Rating Regression (NARRE) [4] improves over this assumption via an attention mechanism which learns a distribution over the individual reviews in the review document. Like DeepCoNN, NARRE also uses TextCNN to discover latent features for each review to predict ratings. Multi-Pointer Co-Attention Networks (MPCN) [40] works on the same principle as NARRE that not every review is equally important. It creates a user/item review sequence instead of a concatenated document and tries to detect the importance dynamically. MPCN proposes a pointer-based review-by-review learning scheme to infer review importance, unlike NARRE's attention weights. Dual Attention Mutual Learning (DAML) [25] DAML utilizes local and mutual attention of CNNs to jointly learn the features of reviews to enhance the interpretability of the proposed DAML model. Like MPCN and NARRE, DAML does not make an independence assumption of reviews.

Although the review-based models show promising results over their non-text-based predecessors, these models are inherently complex. Sachdeva and McAuley [38] show that a comparatively older method, Hidden Factors and Topics (HFT) [29], outperforms the feature-extraction based methods. HFT employs a traditional MF setup, with an additional regularizer modelling the review text corpus likelihood using Latent Dirichlet Allocation (LDA) [1]. However, since all these methods output only ratings, it is not easy to envision the reasons behind their predictions, and thus, they offer little transparency and explainability. Though MPCN, NARRE and DAML provide some extra information about reviews by weighing their importance, the working remains a black box. In their own right, they cannot be used for giving explanations.

2.2 Explainability

Explainability of recommendations can have several forms; we will adhere to generation-based approaches here.

Template-based. This approach first defines some fixed pre-defined explanation sentence templates, and then fills those templates with different words to personalize them. Explicit Factor Model (EFM) [45] generates explainable recommendations by telling the user that "You might be interested in *[feature]*, on which this product performs well/poorly" while maintaining good recommendation performance of Latent Factor Models (LFM). For opinionated text data, [42] introduce a companion learning task of user preference modelling for a recommendation, in parallel with a factorization-based recommendation. The approach explicitly models how a user describes an item's features with latent factors to explain why he/she should pay attention to a particular feature of a recommended item. For example, "We recommend this *[phone]* to you because of its *[high-resolution screen]*". Using Microsoft Concept Graph, DEAML (Deep Explicit Attentive Multi-View Learning) [11] generates template-based explainable recommendations through attentive multi-view learning. FacT [39] builds regression trees on users and items respectively with user-generated reviews and associates a latent profile to each node on the trees to represent users and items. Their approach integrates regression trees to guide LFMs and uses the learnt tree structure to generate template explanations.

The main issue with template-based explanations is that they sound repetitive and hence, may diminish the persuasiveness. Besides, they do not mimic the style in which the user wrote reviews in the past.

Natural Language Generation-based. Most generative models train a language model to provide explainable recommendations. In [7], the authors design a character-level LSTM [15] model that generates text reviews given a combination of the review and ratings score that express opinions about different factors or aspects of an item. NRT [24] leverages gated recurrent units (GRU) [6] to generate tips. According to the predicted ratings, the model can control the generated tips' sentiment, helping users understand the recommended items' critical features. A multi-task recommendation model, MT [28], jointly learns to perform rating prediction and recommendation explanation. The explanation module employs an adversarial sequence to sequence learning technique to generate and discriminate the user and item reviews, motivated by the architecture of GANs [12]. Once trained, the generator generates explanation sentences. Co-Attentive Multi-Task Learning (CAML) [5] has an encoder-selector-decoder architecture inspired by human's information-processing model in cognitive psychology. The co-attention mechanism is the same as in MPCN [40] and their language model consists of GRUs to generate explanations.

A significant weakness that handicaps these models is the time and cost needed to train these models. Training the conditional language model from scratch brings in multiple disadvantages. The models need to be trained for a large amount of time to achieve fluency that suffices, adversely increasing the cost to train them. Moreover, since they do not use any pre-trained language models trained on vast corpus and use only domain-specific data, the quality of the generated reviews is inferior, affecting the explainability.

2.3 Controlled Text Generation

Concerning generating synthetic reviews as explanations, we cannot use language models directly as they do not care about the sentiment (rating). Hence, text generation with respect to a controlling attribute (sentiment) is needed in this case [16, 19, 27]. Also, though [35] does not specifically talk about explainable recommendations, it does indeed answer the same research question for the generative part of the model. These approaches train the language model from scratch, which requires enormous computational power and training data. Moreover, training the language model on a vast corpus may lead to generated reviews not being domain-specific, *i.e.*, they may not adhere to the dataset at hand specifically. The main aim of these models is to generate fluent sentences while simultaneously satisfying the control attribute passed to them. Though they generate very high-quality sentences, they are very complex to implement and have many parameters (CTRL [19] has 1.63 billion parameters!). Training of such models is computationally expensive and not cost-effective due to the usage of multiple GPUs, limiting their usage for domain-specific explainability. The proposed framework, ReXPlug, employs the Plug and Play Language Model (PPLM) [8] that works on top of a pre-trained mammoth language model like GPT-2 [36] for the generation. For generating explanations, an easily trainable attribute model (discriminator) steers the pre-trained language model to generate reviews satisfying the input rating (predicted). In doing so, the combined effect is that the mammoth language model ensures fluency of the generated review, while the discriminator satisfies the predicted rating. This makes ReXPlug very efficient and effortless to train by utilizing the power of transfer learning.

3 THE PROPOSED MODEL

Problem Formulation. Let $r_{(u,i)} \in \mathbb{R}_{>0}$ be the positive real-valued rating that exists between user u and item i. Let $\Omega = \{(u,i) :$ user u rates item $i\}$. Here, unavailable ratings are represented by 0, that is, $r_{(u,i)} = 0$ where $(u,i) \notin \Omega$. In addition, we have user-given natural language reviews associated with each available rating $r_{(u,i)}, \forall (u,i) \in \Omega$. We use $\mathcal{D}_u$ and $\mathcal{D}_i$ to denote reviews given by user u and received by item i, respectively. Given partially available ratings $r_{(u,i)}, \forall (u,i) \in \Omega$, and associated natural language reviews $\mathcal{D}_u, \forall u$ and $\mathcal{D}_i, \forall i$, our aim is to predict unavailable ratings $r_{(u,i)}, \forall (u,i) \notin \Omega$, and more importantly, generate explanations to justify the predicted ratings. This section explains the modules that combine to form the proposed ReXPlug framework for explainable recommendations. ReXPlug consists of three modules as shown in Figure 1: (a) a neural network which leverages user reviews for regularization for improving rating predictions, (b) a cross-attention network which learns to identify the usefulness of each user and item reviews, and (c) an efficient transfer learning-based controlled text generator. We describe these modules in detail now.

3.1 Rating Prediction

To generate explainable recommendations, we first build a simple neural network and a regularizing network that uses the reviews as a regularizer to learn user and item embeddings. The regularization ensures that both ratings and reviews influence the embeddings. This leads to better user behaviour and item properties being captured in the embedding vectors. We set up two parallel feed-forward neural networks, sharing the same trainable user and item embeddings as inputs ((a) in Figure 1). The first feed-forward neural network ($\mathcal{F}_P(\cdot)$) does regression over the ratings, while the second parallel feed-forward neural network ($\mathcal{F}_{RR}(\cdot)$) acts as a regularizer network. The regularizer takes in the user and item embeddings as inputs and predicts the corresponding vectorized review text. We term this construction as Review Regularization (RR).

To represent the review text as a fixed-dimensional vector, we leverage the Deep Averaging Network [17] based Universal Sentence Encoder [3]. From this, we get a fixed-dimensional (512-dimensional) vector. These vectors are treated as targets for the regularizing model, which outputs 512-dimensional vectors and tries to make the predictions as close as possible to the encoded reviews. Note that the regularizer network is active only during the training phase and disabled during the test phase. We formally describe the above procedure as follows.

Let $P \in \mathbb{R}^{d \times m}$ and $Q \in \mathbb{R}^{d \times n}$ be user and item embedding matrices for the users and items. Let $x_u \in \mathbb{R}^m$ and $y_i \in \mathbb{R}^n$ be one-hot encoding vectors for user u and item i. We obtain user and item embedding vectors as follows:

$$p_u = P x_u, \text{ and } q_i = Q y_i \tag{1}$$

where p_u and q_i ($\in \mathbb{R}^d$) are embedding vectors for user u and item i, and m and n denote the number of users and items respectively. These embedding vectors are then concatenated and passed through a feed-forward neural network $\mathcal{F}_P(\cdot)$. This provides us a representation for the user-item pair, (u,i):

$$\begin{aligned} \psi_{(u,i)} &= \mathcal{F}_P(\phi_{(u,i)}), \qquad (2)\\ &= a(W_L(a(W_{L-1}(\ldots a(W_1\phi_{(u,i)} + b_1)\ldots) + b_{L-1})) + b_L), \end{aligned}$$

where $\phi_{(u,i)} = p_u \| q_i$. Here, $\psi_{(u,i)} \in \mathcal{R}^{d'}$ denotes user-item interaction representation, $\|$ denotes concatenation operation and $\phi_{(u,i)}$ is obtained by concatenating the embeddings of user u and item i. Let $w \in \mathbb{R}^{d'}$ be a weight vector. We predict user u's rating on item i as follows:

$$\hat{r}_{(u,i)} = w^T \psi_{(u,i)} \tag{3}$$

where $\hat{r}_{(u,i)}$ denotes predicted rating. The loss function corresponding to rating prediction is given as follows:

$$\mathcal{L}_P(\Theta) = \frac{1}{|\Omega|} \sum_{(u,i)\in\Omega} (r_{(u,i)} - \hat{r}_{(u,i)})^2, \tag{4}$$

where $\Omega = \{(u,i) :$ user u interacts with item $i\}$.

Review Regularizer: We exploit the textual reviews given by the users to regularize $\phi_{(u,i)}$. During the training time, with the help of Universal Sentence Encoder [3], we first get a fixed-sized vector for the review associated with user u and item i. Let this encoded review be denoted by $s_{(u,i)}$, having dimension d_e. Second, we pass $\phi_{(u,i)}$ through a feed-forward neural network $\mathcal{F}_{RR}(\cdot)$ and obtain $\hat{s}_{(u,i)_{RR}}$. Also, we extract the corresponding review for the user-item pair (u,i), denoted by $s_{(u,i)}$. We construct a loss function $\mathcal{L}_{RR}(\cdot)$ between $s_{(u,i)}$ and $\hat{s}_{(u,i)_{RR}}$. This is then used for regularizing $\phi_{(u,i)}$. The loss function corresponding to review regularization is given

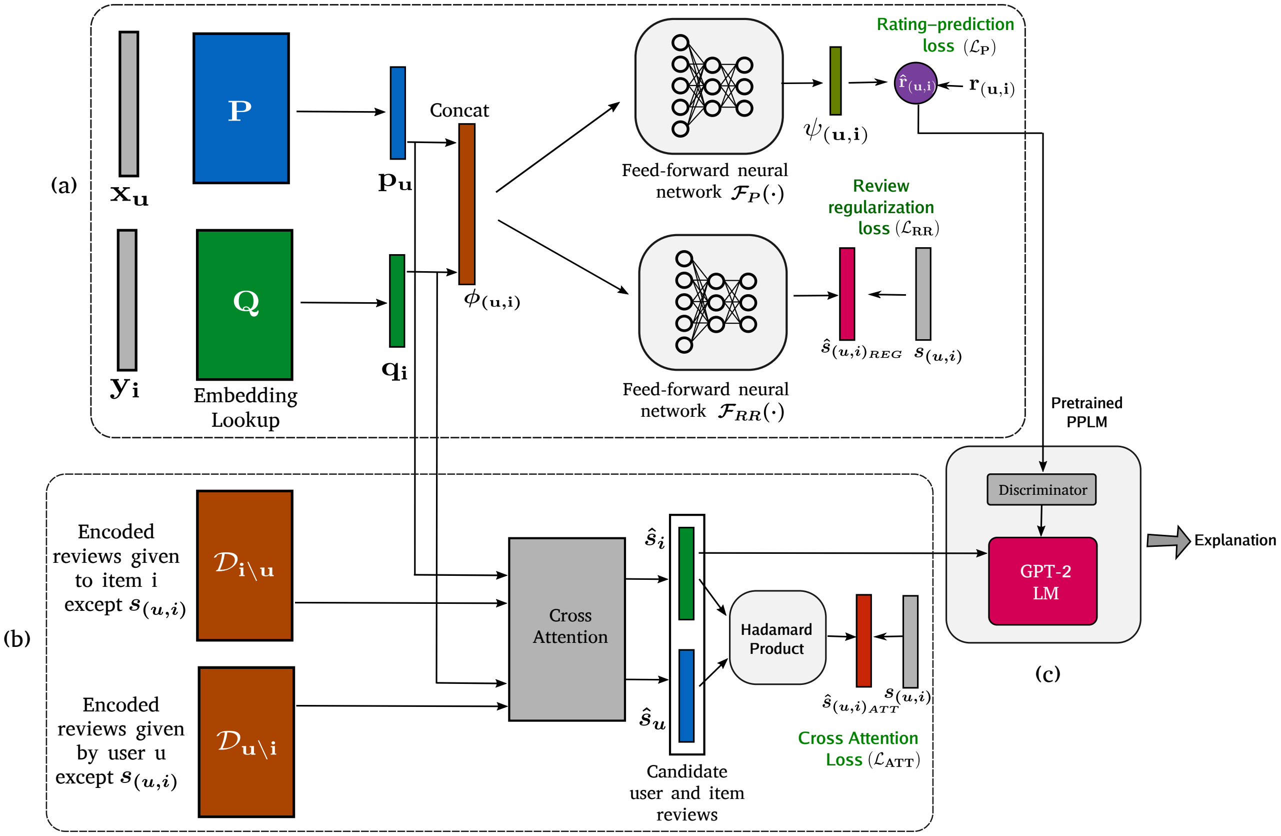
(a)
$\mathbf{x_u}$
$\mathbf{y_i}$
P
Q
Embedding Lookup
$\mathbf{p_u}$
$\mathbf{q_i}$
Concat
$\phi_{(\mathbf{u,i})}$
Feed-forward neural network $\mathcal{F}_P(\cdot)$
$\psi_{(\mathbf{u,i})}$
Rating–prediction loss ($\mathcal{L}_\mathrm{P}$)
$\hat{\mathbf{r}}_{(\mathbf{u,i})}$
$\mathbf{r}_{(\mathbf{u,i})}$
Review regularization loss ($\mathcal{L}_\mathrm{RR}$)
Feed-forward neural network $\mathcal{F}_{RR}(\cdot)$
$\hat{s}_{(u,i)REG}$
$s_{(u,i)}$
Pretrained PPLM
Discriminator
GPT-2 LM
Explanation
(c)
(b)
Encoded reviews given to item i except $s_{(u,i)}$
$\mathcal{D}_{\mathbf{i}\backslash\mathbf{u}}$
Encoded reviews given by user u except $s_{(u,i)}$
$\mathcal{D}_{\mathbf{u}\backslash\mathbf{i}}$
Cross Attention
$\hat{s}_i$
$\hat{s}_u$
Candidate user and item reviews
Hadamard Product
$\hat{s}_{(u,i)ATT}$
$s_{(u,i)}$
Cross Attention Loss ($\mathcal{L}_\mathrm{ATT}$)

Figure 1: The illustration of ReXPlug Architecture. (a), by oneself, is termed as the Review Regularizer (RR) network and (a) and (b) together are called Review Regularizer with Cross-Attention (RRCA). RRCA predicts the rating, $\hat{r}_{(u,i)}$ for a user-item pair (u, i). (b) also gives pointers to the most relevant user and item reviews. $\hat{r}_{(u,i)}$ from (a) and the most relevant item review from (b) are given to pre-trained (c) to generate explanation for (u, i) pair.

as follows:

$$\mathcal{L}_{RR}(\Theta) = \frac{1}{|\Omega|} \sum_{(u,i)\in\Omega} \left\| s_{(u,i)} - \hat{s}_{(u,i)_{RR}} \right\|^2 . \qquad (5)$$

3.2 Cross-Attention (CA)

Next, we construct an attention-based parallel network, termed as the Cross-Attention (CA) network. We have a list of encoded reviews written by user u, given by D_u, encoded as earlier with the Universal Sentence Encoder. Similarly, each item i receives a list of reviews; each encoded similarly to form D_i. Let n_u and n_i denote the number of reviews associated with user u and item i. The task of this network is to find out the one most *relevant* user and item reviews for a user-item pair, which influences the final predicted rating, $\hat{r}_{(u,i)}$. The importance of this network is two-fold: generate better user and item embeddings by identifying the reviews which influence the rating the most; and during test time, feed those identified reviews to the PPLM as conditional text. We show the effectiveness of this network using ablation study in the next section. As a whole, we collectively term the three networks as the Review Regularizer with Cross-Attention (RRCA) network. Figure 2 visually illustrates the cross-attention network and Figure 1 demonstrates how it works in tandem with rating prediction and review generation modules.

During the training time, we first mask out the ground truth review vector, $s_{(u,i)}$, from D_u and D_i. Let the resultant lists be denoted by $D_{u\backslash i}$ and $D_{i\backslash u}$, respectively. The number of reviews in $D_{u\backslash i}$ will thus be $n'_u = n_u - 1$. The task of this network is to find out a pointer to a review vector from each of $D_{u\backslash i}$ and $D_{i\backslash u}$ that are semantically closest to the masked out review, $s_{(u,i)}$. Let these encoded reviews be represented as $\hat{s}_u$ and $\hat{s}_i$, respectively. Like in the regularizer network, the target here is the vector, $s_{(u,i)}$. To find out the *relevant user review vector* with an affinity towards the item i, we treat the item embedding q_i as the query vector. The *key* and *value* vectors are the encoded reviews in $D_{u\backslash i}$. Note that

$$q_i \in \mathbb{R}^{d_e \times n_q}, \text{ and } D_{u\backslash i} \in \mathbb{R}^{n'_u \times d_e}, \qquad (6)$$

where $n_q = 1$ and n'_u is the number of reviews written by user u - 1 (because of masking $s_{(u,i)}$). The cross-attention mechanism computes the inner product of q_i with $D_{u\backslash i}$ and applies *Softmax* on this energy vector to give us a probability distribution, Π_u, over

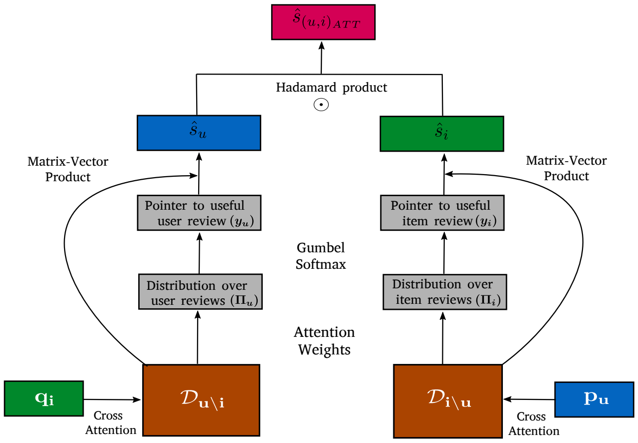
$\hat{s}_{(u,i)_{ATT}}$
Hadamard product
$\odot$
$\hat{s}_u$
$\hat{s}_i$
Matrix-Vector Product
Matrix-Vector Product
Pointer to useful user review (y_u)
Pointer to useful item review (y_i)
Gumbel Softmax
Distribution over user reviews (Π_u)
Distribution over item reviews (Π_i)
Attention Weights
$\mathbf{q_i}$
$\mathcal{D}_{\mathbf{u\setminus i}}$
$\mathcal{D}_{\mathbf{i\setminus u}}$
$\mathbf{p_u}$
Cross Attention
Cross Attention

Figure 2: Illustration of the Cross-Attention (CA) Network during training. Embeddings p_u and q_i attend over $D_{i\setminus u}$ and $D_{u\setminus i}$, respectively. After computing attention weights, Π_u and Π_i, CA applies Gumbel-Softmax to the distributions to get near-one-hot distributions as pointers to the most relevant user and item reviews.

user u's reviews.

$$\Pi_u = \text{softmax}(q_i^T D_{u\setminus i}),$$

$$\text{such that } \Pi_u = [\Pi_u^1, \Pi_u^2, ..., \Pi_u^{n'_u}],$$

$$\sum_{j=1}^{n'_u} \Pi_u^j = 1 \text{ and } \Pi_u^j \geq 1 \forall j \in [n'_u].$$

Now, to find out the pointer to $\hat{s}_u$ which is semantically the closest to $s_{(u,i)}$, thus reflecting the rating, we could have taken the arg max over the obtained attention weights. However, doing this will break differentiability and will stall the training of the model. To get over this, we utilize the *Gumbel-Softmax* trick [18] which gives us a differentiable approximation to arg max. The technique approximates the sampling process of the discrete distribution vector, Π_u. The performance of the network depends on the choice of the temperature parameter τ. Let $\mathcal{G}_j \sim Gumbel(0,1)$, for $j = 1, ..., n'_u$ be i.i.d samples. Each element of the output vector of this stage $y_u \in \mathbb{R}^{n'_u}$ can be obtained as follows:

$$y_u^j = \frac{\exp\left(\frac{\mathcal{G}_j + \log \Pi_u^j}{\tau}\right)}{\sum_{k=1}^{n'_u} \exp\left(\frac{\mathcal{G}_k + \log \Pi_u^k}{\tau}\right)}, \qquad \text{for } j = 1, ..., n'_u. \tag{7}$$

Now, as $\tau \rightarrow 0$, the softmax computation smoothly approaches the arg max, and the output vector y_u approaches one-hot. Then, to get the relevant user vector,

$$\hat{s}_u = y_u^T D_{u\setminus i}. \tag{8}$$

Similarly, we replicate the cross-attention mechanism on the item review vectors, with the query vector being the user embedding q_u, and the encoded reviews in the list $D_{i\setminus u}$ assuming the role of the *key* and *value* vectors. The relevant item vector is $\hat{s}_i = y_i^T D_{i\setminus u}$.

We let the MSE loss function govern the learnings of q_u and q_i through this network during the training time. This module's output vector is the element-wise multiplication of $\hat{s}_u$ and $\hat{s}_i$,

$$\hat{s}_{(u,i)_{ATT}} = \hat{s}_u \odot \hat{s}_i. \tag{9}$$

The masked out review is $s_{(u,i)}$ for the user-item pair, (u,i). If we denote the output for the j^{th} pair as $\hat{s}_{(u,i)_{ATT}}$, the loss of this network is,

$$\mathcal{L}_{ATT}(\Theta) = \frac{1}{|\Omega|} \sum_{(u,i)\in\Omega} \left\| s_{(u,i)} - \hat{s}_{(u,i)_{ATT}} \right\|^2. \tag{10}$$

Empirically, we have observed that this parallel cross-attention network's inclusion leads to a decrease in the MSE, which we show as an ablation study in the next section. Additionally, the cross-attention network serves another vital purpose of fetching the conditional text sentence during test and generation time. These retrieved conditional sentences are then passed as one of the inputs to PPLM for controlled text generation. Thus, the RRCA jointly learns the user and item embeddings from 3 parallel networks: from the direct user-item interactions through the predictor network $\mathcal{L}_P$; from the review regularizer network $\mathcal{L}_{RR}$; and from the cross-attention network. This leads to the generation of robust embeddings for each user and item.

The overall loss during the training time is a linear combination of the losses of the two networks, *i.e.*,

$$\mathcal{L} = \mathcal{L}_P + \lambda_{RR} * \mathcal{L}_{RR} + \lambda_{ATT} * \mathcal{L}_{ATT}, \tag{11}$$

where λ_{RR} and λ_{ATT} are positive real-valued hyperparameters.

3.3 Generating Controlled Text Reviews:

In Figure 1, (c) represents the generative part of ReXPlug. We first train the attribute model (discriminator) of the PPLM. This discriminator is a sentiment classifier that takes a textual sentence as input, and predicts the corresponding sentiment class, with the sentiments being the ratings. We use the same dataset as used in the first module so that the discriminator gets tuned to sync with the dataset's vocabulary. This personalizes the generation process to produce more domain-specific reviews. Training the discriminator is a relatively computationally inexpensive task against the other controlled text generation models [16, 19, 27]. In order to control the output of the language model, at every generation step t, PPLM shifts the history H_t in the direction of the sum of two gradients: one toward higher log-likelihood (LL) of the attribute a under the conditional attribute model $p(a|x)$ and one toward higher LL of the unmodified language model $p(x)$. Combining these factors with a multiplier α provides us with a controllable "knob" to guide generation in a given direction with specified strength [8].

ReXPlug's explanation generator, based on PPLM, accepts two inputs to generate synthetic reviews using the trained discriminator: the rating predicted, $\hat{r}_{(u,i)}$, by $\mathcal{F}_P(\cdot)$ as a sentiment; and the conditional text from the cross-attention network. It then generates multiple candidate sample reviews which entail the conditional text and fulfil the input sentiment. During the test time, the predictor module predicts the rating $\hat{r}_{(u,i)}$ for an unseen pair. The cross-attention network, which, after training, learns to identify the reviews which satisfy the predicted rating, gives us two candidate reviews. We pass the candidate item review and $\hat{r}_{(u,i)}$ as inputs

Table 2: Data Statistics (left) and MSE values (right) of various models. Bold-faced values represent the best model in that row. RRCA and RR are modules of the proposed ReXPlug framework.

Dataset	Data Statistics				Mean Squared Error					
	Reviews	Users	Items	Density	RRCA (Ours)	RR(Ours)	DeepCoNN [46]	NARRE [4]	MPCN [40]	NeuMF [14]
Amazon Digital Music	64K	5K	3K	0.3273 %	**0.6693 ± 0.05**	0.7722 ± 0.08	0.8376 ± 0.01	1.3854 ± 0.35	0.9063 ± 0.02	1.0213 ± 0.18
Amazon Video Games	0.23M	24K	10K	0.0894 %	**1.0617 ± 0.05**	1.0938 ± 0.05	1.1366 ± 0.02	1.1479 ± 0.02	1.2842 ± 0.02	1.1178 ± 0.06
Amazon Clothing	0.27M	39K	23K	0.0307 %	1.0538 ± 0.10	1.0647 ± 0.11	1.1407 ± 0.01	**1.1406 ± 0.01**	1.1637 ± 0.01	1.2343 ± 0.10
Yelp Subset 1	0.99M	83K	29K	0.0411 %	**1.2729 ± 0.12**	1.3032 ± 0.10	1.4096 ± 0.01	1.4454 ± 0.02	1.6151 ± 0.01	1.4261 ± 0.15
Yelp Subset 2	1.04M	87K	30K	0.0403 %	**1.2335 ± 0.16**	1.2599 ± 0.18	1.4047 ± 0.01	1.4339 ± 0.02	1.6051 ± 0.02	1.3842 ± 0.28
BeerAdvocate	1.47M	15K	22K	0.4566 %	**0.3053 ± 0.04**	0.3101 ± 0.04	0.3762 ± 0.01	0.4110 ± 0.10	0.4287 ± 0.01	0.3385 ± 0.03

to the PPLM. The form of the model is based on the Bayes' law: $p(x|a) \propto p(x)p(a|x)$, where a represents the attribute (rating) and x represents the output of the language model.

As we show later, qualitative and quantitative analyses suggest that using the identified item review (encoded as $\hat{s}_i$) performs better than the identified user review. This result can be attributed to the fact that user u's identified review need not specifically talk about item i, leading to irrelevant explanations. In contrast, $\hat{s}_i$ is the encoded identified item review, which is semantically the closest to user u's embedding q_u. Hence, choosing the identified item review also accommodates the user's behaviours.

4 EXPERIMENTS

This section details the experimental settings and compares the results obtained with other state-of-the-art mechanisms. We also conduct some ablation studies to verify the effectiveness of the models. Lastly, we do several qualitative and quantitative analyses of the generated reviews.

4.1 Datasets Used

We use datasets that consist of user-item interactions with reviews for building ReXPlug and other baseline models. We perform our experiments on six datasets, covering a wide range of product review spectrum. All of these datasets are 5-core in nature, *i.e.*, each user has written, and each item has received at least five reviews. Table 2 logs the statistics of these datasets. We also report the data density corresponding to each dataset.

Amazon: Three of these are Amazon datasets, namely Digital Music, Clothing, and Video Games. The 5-core versions of these datasets are already available [13] [31].

Yelp: Since the raw Yelp dataset[2] has a very large number of user-item interactions (close to 8 million), to save the computational costs, we have created two disjoint subsets from it, such that the subsets do not have any items in common, and hence, no common interactions between them. Each of these subsets has close to 1 million interactions.

BeerAdvocate: We have used the BeerAdvocate dataset [32] [30], which consists of beer reviews. The 5-core version of this dataset has around 1.47 million interactions.

4.2 Experimental Settings

Embedding sizes: We tried a range of values for the user and item embedding size, d_e, starting from a low value of 8 to an upper limit of 256. Embedding size in the range of 48-72 gave us the best results on the validation sets of all the six datasets.

$\mathcal{F}_P$ Settings: The rating predictor performs well when the number of layers of $\mathcal{F}_P$ is 3. Also, we have used ReLU activations and Dropouts between each layer, with zeroing probability of 0.5.

RR Settings: We observed that even with RR's simple setting, the regularizer easily overfits without learning anything meaningful, and hence, does not contribute much to rating prediction. To impose restrictions, we add Dropout to the linear layer's output with the probability of an element to be zeroed equal to 0.7.

CA Settings: We try to keep the parameters in this module to a bare minimum, as shown in Figure 2. The transformation layer is the only parametric layer of this network. This is done to avoid a reduction in the influence of the attention mechanism.

Epochs and learning rate: We found that a learning rate of 0.002 best suits the training of both RRCA and RR. Further, we observed that the training saturated before 30 epochs for RRCA and 60 epochs for RR.

Weight decay and Normalizing constants: A weight decay of 10^{-5} is also added to prevent overfitting. Further, λ_{ATT} and λ_{RR} are the normalizing constants for the CA and RR networks, respectively. Empirically, they are both set to $\lambda_{ATT} = \lambda_{RR} = 5$.

PPLM:. For training the discriminator, setting the learning rate to 10^{-3} results in good classifications in just two epochs. During generation, we use the recommended hyperparameter values suggested in [8]. We keep the step size, α, for attribute control at 0.2, KL Divergence (λ_{KL}) of 0.01, Post-norm fusion (γ_{gm}) of 0.875 and number of iterations in {1, 2} for efficient and fluent explanations.

4.3 Evaluation Criteria

We consider the mean squared error (MSE) as an evaluation metric for rating prediction, as has been used by many of ReXPlug's predecessors [4, 5, 40, 46]. We compare ReXPlug's RRCA with four baseline models: NeuMF, which uses only ratings, and DeepCoNN, MPCN and NARRE, which use reviews as features to predict ratings. To understand the cross-attention network's effectiveness, we

[2] https://www.yelp.com/dataset/

Table 3: Qualitative Analysis of generated reviews. Regular typeface indicates Cross-Attention network's identified candidate item review. Bold section indicates ReXPlug's appended explanation.

Ground Truth Review	Generated Review
Love this band. This song is just one more good song in a long list. I would recommend it and its album.	Really like this band. Good music. **I will recommend this to anyone who is a fan of their music. Get this immediately. This album is worth buying!**
WONDERFUL CAFE! Staff is very friendly and the food is amazing! Great place to stop in for lunch! Highly recommend!	Consistently great food, friendly service, and an all-around hidden gem! **I'm surely visiting this place again for lunch or dinner! Hope to see more of this place in my future visits.**
A beautiful collection of Alison Krauss. If you only have one CD of hers...this is the one to have! Absolutely Beautiful!	As always, Alison is amazing. I love this CD better than all the others I have. **It is a must have for anyone who loves music. I'm sure there will be many people like me who enjoy her play the piano and sing and they'll completely love it.**
I bought these for our niece who just had her first baby. She loved them. I will purchase them as gifts again.	This was for a friends lil baby girl and she said that they love the socks. I would buy from this company again. **Don't waste your money buying other ones, these are much more affordable and look amazing too.**
A very very competent Burton-on-Trent lookalike. Stacks of gypsium, smokey stone, talcium powder. Lots of character yet so subtle. Massively drinkable and very enjoyable. Could be misplaced as an English *bitter* and close the best of its kind in Australia.	A very very competent Burton-on-Trent lookalike. Stacks of gypsium, smokey stone, talcium powder. Lots of character yet so subtle. Massively drinkable and very enjoyable. This was previously known as Bronze but rightly marketed as "English Ale". **The name is probably a reference to the "bitter" taste of the brew.**

remove it from RRCA and use only the review regularizer (RR) to predict ratings. Following [5], to automatically evaluate the generated explanations, we compare ReXPlug with two Retrieval methods and a generative method.

Retrieval methods select linguistic explanations from reviews in the training set. ItemRand is a baseline which randomly chooses an item i's review from the training set for a test-pair of (u, i). LexRank [10] is a widely used unsupervised stochastic graph-based method for text summarization. Given all item i's reviews in the training set, LexRank can generate a summary as an explanation, for all users interacting with i. We consider Co-Attentive Multi-Task Learning (CAML) [5] as a baseline generative method. In ReXPlug, we end the generated reviews when there is either a new line (\n) character or an `<|endoftext|>` token.

To compare the ground truth text and the generated text, we use a commonly used measure called BLEU [34] score. Following [33], we use the BLEU-3 and BLEU-4 granularities. We also consider the Distinct-1 and Distinct-2 [22] scores to gauge the diversity of explanations. Distinct-1 and Distinct-2 scores measure unigram and bigram diversity, respectively.

We also compute the Pearson Correlation Coefficient between the ground truth review from the test set of each dataset and the reviews generated by encoding the two texts with a RoBERTa model [26] trained explicitly for the task of semantic similarity, as given by the STS benchmark [37].

We introduce another metric to see how close the generated explanations' sentiments are with ground truth reviews' sentiments by computing the Mean Absolute Error (MAE) between the two. The metric determines how well the sentiments of the generated explanations align with the ground truth text's sentiment. However, user reviews can be noisy, and so, comparing the generated explanations' sentiments against the true ratings may not be a correct metric. Instead, we consider this problem an ordinal classification problem and train an XLNet classifier on each dataset's validation set (as the validation set is oblivious to the generated explanations; the train and test sets are not) [43]. The classifier learns to classify text into one of the five classes corresponding to sentiments (ratings) 1-5. We then send both the generated and the ground truth reviews to this trained XLNet classifier to obtain their respective sentiments. Finally, we compute MAE between them. An MAE of 1 made by an explanation generating model on the XLNet classifier indicates that the generated explanation's sentiment is within a ±1 range from the ground truth review's sentiment.

4.4 Performance Comparisons of RRCA

To compare RRCA with the models available in the literature, we created five independent random splits with proportions $80 : 10 : 10$ (train/validation/test sets) for each of the six datasets. The MSE averaged across the five splits is reported in Table 2. It shows how well ReXPlug's RRCA and RR perform with respect to many competitive models in the literature for the task of rating prediction. Especially on the Amazon Digital Music dataset, RRCA obtains an improvement of $\sim 14\%$ on MSE. Lower MSE indicates that the quality of recommendations produced by RRCA is very high.

RR is a simple and much more computationally efficient model to train than the comparison models and even RRCA. RR obtains results better than its counterparts across most of the datasets (except for RRCA). Hence, if only recommendations are to be made, RR is a very compelling choice.

A key observation to note is that although RRCA and RR beat these models on rating prediction, they have a higher variance.

Table 4: Quantitative Analysis of generated reviews. Bold-faced values indicate the best method on a dataset. BLEU scores measure similarity between the ground truth text and generated review at various n-grams, while Distinct-n score measures the diversity of generated n-grams. PCC is the Pearson Correlation Coefficient and MAE is the sentiment analysis score.

Dataset	Model	BLEU-3	BLEU-4	Distinct-1	Distinct-2	PCC	MAE
Digital Music	ItemRand	0.1872	0.1173	0.0804	0.3976	0.5201	1.1456
	LexRank	0.2258	0.1406	0.0511	0.3155	**0.6141**	1.0874
	CAML	0.1603	0.0802	0.1703	0.4694	0.4659	0.8697
	ReXPlug	**0.2652**	**0.1519**	**0.1727**	**0.5357**	0.5892	**0.7986**
Video Games	ItemRand	0.1670	0.1246	0.0975	0.3929	0.4458	1.3342
	LexRank	0.2473	0.1538	0.0506	0.2878	0.5141	1.3322
	CAML	0.2718	0.1715	0.1867	0.5324	0.4140	1.2280
	ReXPlug	**0.3878**	**0.2439**	**0.2098**	**0.6066**	**0.5186**	**0.9328**
Clothing	ItemRand	0.2032	**0.1296**	0.1463	0.5441	0.4535	1.2372
	LexRank	0.2111	0.1198	0.1029	0.4662	**0.4898**	1.2518
	CAML	0.0466	0.0282	0.2249	0.5852	0.4542	1.1178
	ReXPlug	**0.2149**	0.1192	**0.2371**	**0.6629**	0.4864	**0.9492**
Yelp 1	ItemRand	0.2668	0.1481	0.0946	0.4178	0.3292	1.4194
	LexRank	0.2694	0.1511	0.0563	0.3369	**0.5255**	1.4424
	CAML	**0.3914**	**0.2249**	0.1721	0.5179	0.4172	1.3377
	ReXPlug	0.3176	0.1994	**0.2476**	**0.6841**	0.5143	**1.1243**
Yelp 2	ItemRand	0.2674	0.1493	0.0922	0.4133	0.3269	1.4281
	LexRank	0.2660	0.1499	0.0566	0.3373	**0.5306**	1.4430
	CAML	**0.3847**	**0.2047**	0.1721	0.5175	0.4157	1.3698
	ReXPlug	0.3307	0.1855	**0.2490**	**0.6861**	0.5177	**1.1419**
BeerAdvocate	ItemRand	0.1253	0.0854	0.0672	0.3671	0.6569	0.5158
	LexRank	0.2991	0.1598	0.0463	0.3256	0.6712	0.5271
	CAML	0.3123	0.1772	**0.1677**	**0.6017**	0.5964	0.4778
	ReXPlug	**0.3208**	**0.1806**	0.0991	0.3686	**0.6742**	**0.4194**

Another observation is that the introduction of the cross-attention network reduces both MSE and variance.

4.5 Analysis of Explanations

Table 3 displays some explanations generated by the ReXPlug framework against the true unseen reviews from the test datasets. As can be seen, the explanations are personalized and carry the same sentiment as the original hidden review. The last example in Table 3 also shows the Cross-Attention network's efficacy. In the BeerAdvocate dataset, a few examples have the same user-item pair, but with slightly different review texts. This duplication corresponds to the fact that the same user has repurchased the same item. The example demonstrates that the cross-attention network can identify the training set reviews semantically closest to the hidden review. It also shows its role in the learning of robust user and item embeddings.

Table 4 quantitatively analyzes the explanations on the evaluation metrics mentioned above. An important observation is that the explanations generated by ReXPlug are very diverse, as is reflected by the Distinct-1 and Distinct-2 scores. Also, the MAE in sentiment analysis is the lowest for ReXPlug across all the datasets. In terms of the BLEU scores, both CAML and ReXPlug give competitive results.

For LexRank, all the users are assigned the same explanations for an item. Because all the six datasets are biased towards the ratings 4 and 5, LexRank's generated summaries and the ground truth reviews primarily have a positive sentiment. The bias leads to LexRank getting a high Pearson Correlation Coefficient. However, it cannot generate diverse text, leading to lower Distinct scores, as is expected. The same is true for ItemRand.

4.6 Ablation Study

Effectiveness of cross-attention. Table 2 shows that RRCA performs better than RR in the task of rating prediction by consistently having a lower MSE across the datasets. The cross-attention module also lowers the variance of RRCA. However, RR is much faster than RRCA. We can view this as a trade-off between time and effectiveness in producing recommendations.

Effectiveness of PPLM. If the predicted rating is close to the actual rating, but the cross-attention network's identified review does not reflect the prediction due to noisy reviews, the conditional text to the PPLM is flawed. However, our empirical observations show that PPLM overturns the conditional text's sentiment based on the input predicted rating. Hence, the generated reviews are corrected even when the cross-attention network fails, making ReXPlug's architecture very robust to noise.

5 CONCLUSION AND FUTURE WORK

This paper proposes a plug and play way of generating an explainable recommendation. The model is extremely computationally efficient compared to the baseline models as it trains in a much lesser time. Experiments show that our approach outperforms state-of-the-art comparison models on the rating prediction task and generates high-quality, fluent, diverse, domain-specific and personalized reviews as explanations. ReXPlug's review generation process during the inference time is, however, comparatively slower. The gradient perturbation by PPLM affects the speed of review generation, making it slower than the LSTM/GRU-based models. Our future work aims to address this concern.

REFERENCES

[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. 3 (March 2003), 993–1022.

[2] Rose Catherine and William W. Cohen. 2017. TransNets: Learning to Transform for Recommendation.. In *RecSys*. ACM, 288–296. http://dblp.uni-trier.de/db/conf/recsys/recsys2017.html#CatherineC17

[3] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Brussels, Belgium, 169–174. https://doi.org/10.18653/v1/D18-2029

[4] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural Attentional Rating Regression with Review-Level Explanations. In *Proceedings of the 2018 World Wide Web Conference* (Lyon, France) *(WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1583–1592. https://doi.org/10.1145/3178876.3186070

[5] Zhongxia Chen, Xiting Wang, Xing Xie, Tong Wu, Guoqing Bu, Yining Wang, and Enhong Chen. 2019. Co-Attentive Multi-Task Learning for Explainable Recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. 2137–2143. https://doi.org/10.24963/ijcai.2019/296

[6] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1724–1734. https://doi.org/10.3115/v1/D14-1179

[7] 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* (Tokyo, Japan) *(IUI '18 Companion)*. Association for Computing Machinery, New York, NY, USA, Article 57, 2 pages. https://doi.org/10.1145/3180308.3180366

[8] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. In *International Conference on Learning Representations*. https://openreview.net/forum?id=H1edEyBKDS

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 4171–4186.

[10] Günes Erkan and Dragomir R. Radev. 2004. LexRank: Graph-Based Lexical Centrality as Salience in Text Summarization. *J. Artif. Int. Res.* 22, 1 (Dec. 2004), 457–479.

[11] Jingyue Gao, Xiting Wang, Yasha Wang, and Xing Xie. 2019. Explainable Recommendation Through Attentive Multi-View Learning. In *AAAI Conference on Artificial Intelligence (AAAI)*. https://www.microsoft.com/en-us/research/publication/explainable-recommendation-through-attentive-multi-view-learning/

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, Vol. 27. Curran Associates, Inc., 2672–2680.

[13] 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*. 507–517.

[14] 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*. 173–182.

[15] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (Nov. 1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[16] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Controllable Text Generation. *CoRR* abs/1703.00955 (2017).

[17] Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep Unordered Composition Rivals Syntactic Methods for Text Classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, 1681–1691. https://doi.org/10.3115/v1/P15-1162

[18] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=rkE3y85ee

[19] Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A Conditional Transformer Language Model for Controllable Generation. abs/1909.05858 (2019).

[20] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1746–1751.

[21] Y. Koren, R. Bell, and C. Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37. https://doi.org/10.1109/MC.2009.263

[22] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 110–119.

[23] 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* (Shinjuku, Tokyo, Japan) *(SIGIR '17)*. Association for Computing Machinery, New York, NY, USA, 345–354. https://doi.org/10.1145/3077136.3080822

[24] 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* (Shinjuku, Tokyo, Japan) *(SIGIR '17)*. Association for Computing Machinery, New York, NY, USA, 345–354. https://doi.org/10.1145/3077136.3080822

[25] Donghua Liu, Jing Li, Bo Du, Jun Chang, and Rong Gao. 2019. *DAML: Dual Attention Mutual Learning between Ratings and Reviews for Item Recommendation*. Association for Computing Machinery, New York, NY, USA, 344–352. https://doi.org/10.1145/3292500.3330906

[26] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019).

[27] Lajanugen Logeswaran, Honglak Lee, and Samy Bengio. 2018. Content preserving text generation with attribute controls. In *Advances in Neural Information Processing Systems*, Vol. 31. Curran Associates, Inc., 5103–5113.

[28] Yichao Lu, Ruihai Dong, and Barry Smyth. 2018. Why I like It: Multi-Task Learning for Recommendation and Explanation. In *Proceedings of the 12th ACM Conference on Recommender Systems* (Vancouver, British Columbia, Canada) *(RecSys '18)*. Association for Computing Machinery, New York, NY, USA, 4–12. https://doi.org/10.1145/3240323.3240365

[29] Julian McAuley and Jure Leskovec. 2013. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys '13)*. Association for Computing Machinery, New York, NY, USA, 165–172.

[30] 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.

[31] 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*. 43–52.

[32] Julian John McAuley and Jure Leskovec. 2013. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *Proceedings of the 22nd international conference on World Wide Web*. 897–908.

[33] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 188–197. https://doi.org/10.18653/v1/D19-1018

[34] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (Philadelphia, Pennsylvania) *(ACL '02)*. Association for Computational Linguistics, USA, 311–318. https://doi.org/10.3115/1073083.1073135

[35] Alec Radford, Rafal Józefowicz, and Ilya Sutskever. 2017. Learning to Generate Reviews and Discovering Sentiment. *CoRR* abs/1704.01444 (2017).

[36] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. (2019).

[37] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 3982–3992. https://doi.org/10.18653/v1/D19-1410

[38] Noveen Sachdeva and Julian McAuley. 2020. How Useful are Reviews for Recommendation? A Critical Review and Potential Improvements. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Jul 2020). https://doi.org/10.1145/3397271.3401281

[39] Yiyi Tao, Yiling Jia, Nan Wang, and Hongning Wang. 2019. The FacT: Taming Latent Factor Models for Explainability with Factorization Trees. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 295–304. https://doi.org/10.1145/3331184.3331244

[40] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Multi-Pointer Co-Attention Networks for Recommendation.. In *KDD*. ACM, 2309–2318. http://dblp.uni-trier.de/db/conf/kdd/kdd2018.html#TayLH18

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc., 5998–6008.

[42] Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. 2018. Explainable Recommendation via Multi-Task Learning in Opinionated Text Data. In *The 41st International ACM SIGIR Conference on Research &#38; Development in Information Retrieval* (Ann Arbor, MI, USA) *(SIGIR '18)*. ACM, 165–174. https://doi.org/10.1145/3209978.3210010

[43] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems*, Vol. 32. 5753–5763.

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

[45] 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* (Gold Coast, Queensland, Australia) *(SIGIR '14)*. Association for Computing Machinery, New York, NY, USA, 83–92. https://doi.org/10.1145/2600428.2609579

[46] Lei Zheng, Vahid Noroozi, and Philip S. Yu. 2017. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining* (Cambridge, United Kingdom) *(WSDM '17)*. Association for Computing Machinery, New York, NY, USA, 425–434.