Factual and Informative Review Generation for Explainable Recommendation

Zhouhang Xie,[1] Sameer Singh,[2] Julian McAuley, [1] Bodhisattwa Prasad Majumder [1]

[1] University of California, San Diego
[2] University of California, Irvine
{zhx022, jmcauley, bmajumde}@ucsd.edu, sameer@uci.edu

Abstract

Recent models can generate fluent and grammatical synthetic reviews while accurately predicting user ratings. The generated reviews, expressing users' estimated opinions towards related products, are often viewed as natural language 'rationales' for the jointly predicted rating. However, previous studies found that existing models often generate repetitive, universally applicable, and generic explanations, resulting in uninformative rationales. Further, our analysis shows that previous models' generated content often contain factual hallucinations. These issues call for novel solutions that could generate both *informative* and *factually grounded* explanations. Inspired by recent success in using retrieved content in addition to parametric knowledge for generation, we propose to augment the generator with a personalized retriever, where the retriever's output serves as external knowledge for enhancing the generator. Experiments on Yelp, TripAdvisor, and Amazon Movie Reviews dataset show our model could generate explanations that more reliably entail existing reviews, are more diverse, and are rated more informative by human evaluators.

1 Introduction

Recently, there has been increasing interest in treating review generation as a proxy for explainable recommendation, where generated reviews serve as rationales for the models' recommendations(Li et al. 2016; Li, Zhang, and Chen 2021; Ni et al. 2017; Ni and McAuley 2018). However, existing models commonly generate repetitive and generic content, resulting in uninformative explanations (Geng et al. 2022). Further, when evaluating the factuality of generated reviews using pre-trained entailment model, our analysis shows that existing models are also susceptible to factual hallucination, a long-existing challenge in many natural language generation (NLG) tasks (Pagnoni, Balachandran, and Tsvetkov 2021; Maynez et al. 2020). Specifically, the models often generate statements that are not supported by information about the corresponding product in the training set. Both nonfactual and uninformative explanations are undesirable, as end users would look for recommendation rationales that truthfully reflect the characteristics of the product without being overly generic. Thus, these problems limit the usability of natural language explanations (NLE) produced by existing explainable recommendation models.

Arxiv preprint, 2022.

In order to address the issue that models commonly generate univiersally correct explanations, previous works experimented with diversifying generated reviews using distantly retrieved images as additional signals (Geng et al. 2022). However, recommender system datasets do not always have associated images, and Geng et al. proposed to retrieve images from the web using available textual data. While this method indeed significantly diversifies the generated natural language explanations (NLE), there is no guarantee that the retrieved content will truthfully represent the quality of the corresponding product. Thus, the generator needs to condition on a given feature or aspect the user cares about at inference time, commonly extracted from the *ground-truth* review. This limits the usability of models as such user input might not be available at inference time. Another line of work attempted to incorporate a pre-trained language model for better generation quality (Li, Zhang, and Chen 2021). However, the same study shows that pre-trained language models such as GPT-2 struggle to produce diverse reviews while maintaining competitive recommendation accuracy. Thus, generating informative and factual reviews without having access to information in ground truth reviews remains an open problem.

Recent advances in knowledge-grounded NLG show that retrieving unstructured text as supplements for the model's parametric knowledge significantly improves factuality and diversity of generated content (Lewis et al. 2020; Guu et al. 2020). Inspired by such success, in this work, we propose to leverage existing reviews as additional context for generating recommendation explanations. Specifically, we propose **P**ersonalized **R**etriever **A**ugmented **G**enerator (**PRAG**), a model loosely based on the retriever-reader framework for explainable recommendation. PRAG consists of a personalized retriever that can accurately give a rating estimation and generate a latent query for review prediction given the input user and item. To encourage *factual* explanations, we formulate the task of NLE generation as question-answering, where a reader model produces the explanation grounded on retrieved content. Meanwhile, to ensure the *informativeness* of the explanations, we estimate a set of personalized high-tf-idf keywords using the latent query, and use these keywords to guide the answer of the reader model. In this way, the reader model produces the final explanation by abstracting input text snippets and keywords, which yields superior

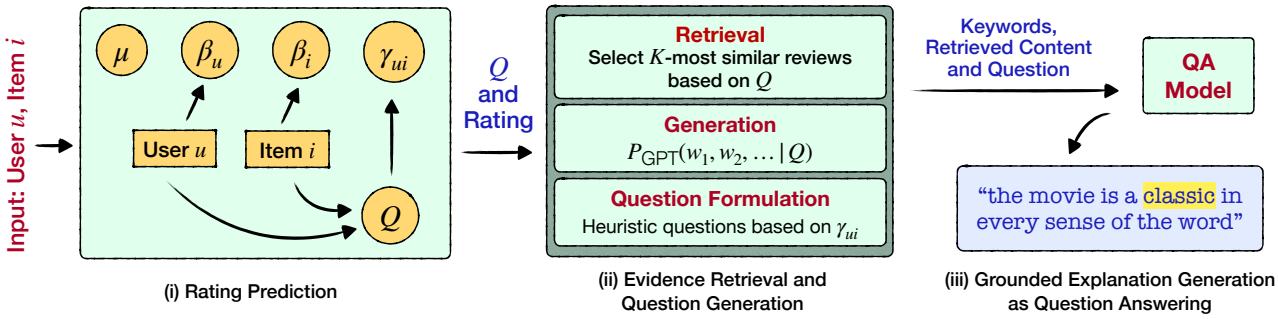

Figure 1: **The proposed framework PRAG**. Given the input user and item, our model produces a rating estimation (Section 4.5), as well as a latent query Q (Section 4.4). We then use the query to retrieve existing reviews as evidence for the recommendation, and probe a question answering model for natural language explanation (Section 5).

quality compared to previous work in both automatic and human evaluation. Our contributions are as follows:

- To our best knowledge, we are the first work that evaluates and highlights the importance of factuality in natural language explanation for recommender systems.
- We develop a novel personalized retriever-reader model for generating factual and informative recommendation explanations.
- We personalize question-answering for generating recommendation explanations.

2 Related Work

Explainable recommendation aims at providing users insight for a recommender systems' decision. Following earlier works that provide topic words and product features as explanations (McAuley and Leskovec 2013; Zhang et al. 2014), recent works are increasingly focusing on generating reviews as explanations (Ni et al. 2017; Ni and McAuley 2018; Hada, Vijaikumar, and Shevade 2021; Dong et al. 2017). However, these works are not focused on the factuality of the generated content. Meanwhile, existing works require training a language model from scratch, while previous attempts at leveraging pre-trained language models either require decoding-time search (Hada, Vijaikumar, and Shevade 2021), or are not as performant as other recommendation models (Li, Zhang, and Chen 2022).

It is to be noted that there are commonly used existing metrics such as Distinct-N (Liu et al. 2022) and unique sentence ratio (USR) (Li, Zhang, and Chen 2020) that focus on diversity of generated explanations. However, a common measure used to increase diversity is to generate a sentence based on words and phrases from the *ground-truth* review, such as in Geng et al. and Ni and McAuley. Our work differs from previous works in that we consider specifically the case where no information from the ground truth review is given, which is common in recommender systems.

Outside of recommender systems, there is a general trend for using natural language as explanations for various tasks, such as text classification (Hancock et al. 2018; Zhou et al. 2020), image captioning (Majumder et al. 2022; Marasović et al. 2020) and question answering (Dalvi et al. 2021; Tafjord and Clark 2021). However, expert annotated explanations are usually unavailable in recommender system datasets, and reviews are usually noisy by nature and require further processing (Bražinskas, Lapata, and Titov 2020, 2021). This leaves learning to generate explanations from noisy supervision an open challenge, which we seek to address in this work.

Another related existing problem in natural language generation tasks is safe and generic outputs. For example, this problem is well studied in dialogue generation, where previous work shows that exposure bias and maximum likelihood objective lead the model to produce universally applicable responses (Zhou, Li, and Li 2021). In another parallel line of work on evaluating natural language explanations, Wiegreffe, Marasović, and Smith pointed out that universally correct explanations are undesirable, calling for diverse and informative explanations. However, these works commonly focus on traditional NLP tasks and well-formulated datasets.

3 PRAG: Setup and Overview

We first introduce the joint review-rating prediction task, then cover our model's general architecture.

3.1 Problem Setup

Consider a set of users U and items I, where each user $u \in U$ and item $i \in I$ in the training set is associated with a real numbered rating $r_{u,i} \in \mathcal{T}$ and a review $e_{u,i} \in E$ that serves as explanations for the users rating decision. The task of joint rating-explanation estimation learns a function $\text{rec} : u, i \rightarrow \hat{r}_{u,i}, \hat{e}_{u,i}$, where $\hat{e}_{u,i}$ is a textual explanation that informs the user of the reasoning behind $\hat{r}_{u,i}$. Note that broadly speaking, explanations can be in various forms, such as topic words and feature importance, here, we use $e_{u,i}$ to denote natural language explanation specifically.

3.2 Model Overview

The architecture of PRAG is depicted in Figure 1. Given the input user u and item i, we first obtain semantic embeddings for all related existing reviews. Here, "related reviews" are reviews written by the user or written about the item. Then,

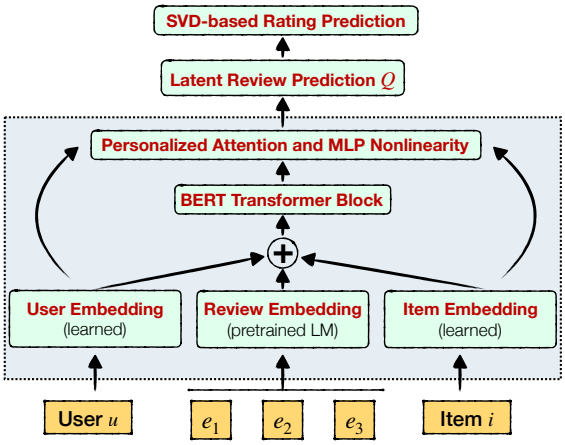

Figure 2: Overview for the **retriever architecture**. Our model takes as input user and item id and a set of corresponding historical reviews, then produces a latent review Q. The latent review is then used for rating prediction, retrieval, and explanation generation.

given the historical reviews, a personalized retriever model produces a latent query $Q_{u,i}$ that is close to the ground truth explanation given the input user and item. A rating prediction module then uses such a latent query to produce a rating estimation.

To generate an explanation, we exploit the rich semantic information in $Q_{u,i}$. Specifically, we (1) retrieve a set of existing reviews G based on $Q_{u,i}$, where $G \subseteq E$, and (2) generate a set of high tf-idf scored keywords that the user would care about based on $Q_{u,i}$ and G. Finally, a keyword-guided question answering model produces the final explanation based on the keywords, retrieved reviews, and the question. We now provide descriptions for each component.

4 Personalized Retriever

The architecture of our retriever model is as shown in Figure 2. The review, item, and user embedding go through a stack of transformer blocks and are then pooled to generate the estimated review embedding.

4.1 Embedding Reviews

Pre-trained language model representations contain rich semantic information about sentences and supports arithmetic operations such as addition and dot product. Following previous success in pre-computing embeddings for efficient retrieval, such as in RAG (Lewis et al. 2020), we obtain sentence embeddings for all reviews using a pre-trained sentence encoder Φ that outputs d dimensional sentence representations. In practice, we adopt the MPNET (Song et al. 2020) model from huggingface transformers (Wolf et al. 2020) for balanced efficiency and embedding quality. We note any state-of-the-art general-purpose sentence encoding model can be used for PRAG, and provide additional analysis with T5 (Raffel et al. 2020) sentence encoders in our experiment section.

4.2 Review Aggregation

Our base personalized retriever model is based on BERT architecture (Devlin et al. 2019). Specifically, we treat each historical review in the training corpus as a token. When predicting the review a user would give to an item, the input to the model is a set of historical reviews written by the user and written for the item. Since these tokens do not represent consecutive words in a sentence as in the original BERT model, we do not use any position embedding. However, historical reviews play different roles when they are related to the item versus being related to the user, thus we choose to maintain learnable embedding $c \in \mathbb{R}^d$ to represent these two distinct scenarios, which are added to the original review embedding instead of position embeddings.

Finally, we maintain a learnable embedding $v_u \in \mathbb{R}^d$ and $v_i \in \mathbb{R}^d$ for each user and item to model user preferences, following common practice in recommender systems. For the input user and item, we look up the corresponding embedding and add these embeddings to each of the input review embedding. The final input review embedding to the model is thus:

$$v_u + v_i + c.$$

Such a final embedding is then passed through a stack of 2-layer transformers to process the cross-review relationship in the input.

4.3 Personalized Attention

Not every review in the input is important. Thus, it is crucial to select reviews that are helpful to the recommendation model. To achieve this, we develop a personalized attention module for weighing review representations from the base transformer model. Specifically, we obtain an attention score with respect to each piece of input reviews using a standard linear layer with a Rectified Linear Unit (ReLU) activation function using the concatenation of the review embedding, user embedding, and item embedding. Following previous works' insight that recommendation models benefit from un-smoothed attention scores' ability to discard irrelevant items (Zhou et al. 2018), we conduct weighted pooling directly using the normalized attention scores.

4.4 Review Embedding Prediction

Finally, the weighted sum of review embeddings is passed through a multi-layer-perception (MLP) layer to produce the final latent query $Q_{u,i} \in \mathbb{R}^d$. During training, we minimize the L2 distance between the produced query and the embedding of the corresponding ground truth review, $\mathcal{L}_{\text{retrieve}}$, following previous success in using vectorized reviews to regularize recommender models (Hada, Vijaikumar, and Shevade 2021). Such embedding thus represents the predicted semantics of the input user and item, which could then be used to retrieve relevant reviews that are semantically similar to the predicted review from the existing review corpus.

4.5 Rating Prediction

To perform rating prediction, we combine HFT (McAuley and Leskovec 2013), a strong matrix-factorization based explainable recommendation model with a modified wide-and-

deep (Cheng et al. 2016) architecture. Specifically, the original HFT model makes predictions by modifying the following equation:

$$\text{rec}(u, i) = \gamma_u \times \gamma_i + \beta_u + \beta_i + \mu,$$

where μ is set to the global mean value of all ratings, and β_u, β_i are the learned bias. The model further ties either γ_u or γ_u to topic models, and learns product or user-specific topics by jointly minimizing the rating regression loss and the negative likelihood of the corresponding review.

We extend the HFT model by using latent query $Q_{u,i}$. Specifically, we adopt the estimated latent review as a new source of semantic information in place of topic models. For example, when associating item features to semantic information, we use a simple multi-layer-perception (MLP) to map $Q_{u,i}$ to γ_i (or γ_u). Meanwhile, previous work has shown that using a shallow linear (wide) layer for memorizing simple patterns could increase the performance of recommendation models (Cheng et al. 2016). Following this intuition, we add an additional linear (wide) layer to the original model

$$\text{rec}(u, i) = \text{MLP}(Q) \times \gamma_i + \text{wide}(Q) + \beta_u + \beta_i + \mu.$$

We learn to predict rating using the standard squared loss $\mathcal{L}_{\text{rating}}$. The final joint loss for training the personalized retriever is then

$$\mathcal{L}_{\text{retrieve}} + \mathcal{L}_{\text{rating}}.$$

5 Explanation Generation as Keyword-guided Question Answering

To generate an explanation, we source information from retrieved reviews and treat the task of explanation generation as question answering. Specifically, the reader model should be able to answer why a higher (or lower) rating *adjustment score* γ_{ui} is being produced, as this is the only inter-user-item factor in rating prediction. To achieve this, we first train an embedding estimator that generates *informative* keywords using the mean embedding of latent query $Q_{u,i}$ as well as its corresponding set of retrieved reviews. After this, we probe a question-answering model trained to *factually* reflect content from the input with natural language prompts while incorporating the generated keywords. The schema of the explanation generation pipeline is as shown in Figure 3.

To facilitate factual behavior for both the embedding estimator and the question-answering model, we design an aggregated task where the ground-truth is *guaranteed* to have a strong correlation to model input at training time. Concretely, we train a model to recover informative keywords in a review from the latent vector encoded from that specific review. Now we discuss each of the components in detail.

5.1 Retrieving Reviews

Given that the latent query is optimized to be similar to the ground truth review, a natural scheme for review retrieval is to rank existing reviews' semantic embedding with respect to $Q_{u,i}$ using similarity metrics. However, in practice, we found the model often produces overly-generic retrieval results, i.e., the reviews for multiple users tend to be similar, as observed in previous work (Geng et al. 2022). To address this issue, we propose to retrieve existing reviews using characteristics that are *specific* to the user. In other words, we want to de-emphasize explanations that a model will produce for *every* user based on the item. We do so by examining the explanation for other users, estimating the latent explanation that are produced for all users, and marginalizing out such a universal explanation.

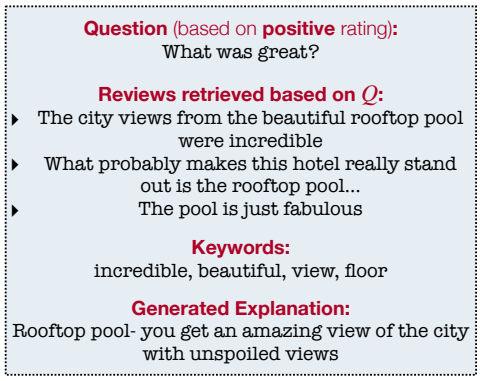

Figure 3: **PRAG** use retrieved historical reviews as source text, and **generates** the explanation based on the keywords.

Concretely, we sample a batch of users at inference time, and obtain each sampled user's corresponding latent query $Q_{u_n,i}$. Then, we estimate the explanation that would be produced for every user using the mean embedding of the batch of predicted queries. Finally, we subtract such mean embedding from the original $Q_{u,i}$, effectively marginalizing out the user-agnostic aspect of the latent query. Since this changes the magnitude of $Q_{u,i}$, we rank existing reviews using cosine distance instead of L2 distance, and select top reviews as the retrieved content. Note that if $Q_{u,i}$ is tied to γ_i, we marginalize out the universal explanation for a batch of items instead, as we need to de-emphasize the explanation being produced for every item. We provide qualitative and quantitative analysis of such marginalization in Section 8.

5.2 Informative Keyword Generation

Retrieved reviews are by nature noisy, and multiple aspects in the reviews could simultaneously contribute to the model's rating decision. However, to generate realistic reviews, the generator must select a few aspects to focus on, as natural reviews typically focus on a few central topic words. To ensure the informativeness of the generated review, we design an embedding estimator that generates high tf-idf scored keywords given the personalized latent query. We then use the keywords in the explanation generation phase for improved informativeness.

To achieve this, we adopt a pre-trained GPT-2 model[1], and formulate the optimization objective as the task of embedding-to-keyword prediction. Specifically, given a latent embedding $\Phi(e)$ encoded from an original review e in the training set, we train the embedding estimator on recov-

[1]https://huggingface.co/distilgpt2

ering the 5-highest tf-idf scored words from the original review. In practice, we use the concatenated target keywords as desired model output and fine-tune the language model using the standard MLE objective. At inference time, we could then condition the embedding estimator on the corresponding latent query since the query is optimized to be in the semantic space of the sentence encoder Φ.

5.3 Explanation Generation

For explanation generation, we adopt a general purpose question-answering model[2] , and exploit the in-context learning ability of large-scale, pre-trained language models (Brown et al. 2020). Specifically, we ask the model to answer the question "`what was great?`" or "`what was not good?`", depending on the sign of the predicted rating adjustment score. Meanwhile, we note that the choice of prompt does not significantly influence the generation quality, and provide experiment with alternative prompts in the appendix.

To guide the model to cover the aspect that the user cares about, we want to encourage the model's output to contain at least one of the keywords produced by the embedding estimator. To achieve this, we first apply constrained decoding (Hokamp and Liu 2017) using the original, unguided question answering model, forcing the output to include at least one of the input keywords. However, this could result in ungrammatical text. To address this issue, we manually rephrase 100 outputs for each dataset, which is easily achievable. This results in a high-quality dataset where each input text is paired with a set of keywords, with a ground-truth answer that contains at least one of the keywords. We then fine-tune the original question answering model on this dataset for a maximum of 10 epochs, following previous work on few-shot question answering (Ram et al. 2021). The fine-tuned model could then be applied for efficient explanation generation *without* any decoding-time constraints.

6 Experiments

6.1 Datasets

We conduct our experiments on three publicly available dataset and splits from various domains (Li, Zhang, and Chen 2020): Yelp[3] (restaurant reviews), TripAdvisor[4] (hotel) and Amazon Movies and TV (He and McAuley 2016). Note that the data splits guarantee that products in the test set always appear in the training set.

6.2 Baselines

We compare four commonly used models in the literature: Att2Seq (Dong et al. 2017), NRT (Li et al. 2017), PETER (Li, Zhang, and Chen 2021), and PEPLER (Li, Zhang, and Chen 2022). Among the models, Att2Seq and NRT are LSTM-based models, while PETER is a transformer-based model. We additionally incorporate a variant of PETER that conditions on a topic word from ground-truth review at inference time, denoted by PETER+. Finally, PEPLER adopts a pre-trained GPT-2 model with prompt tuning for explanation generation. We provide more details for each of the models in the appendix.

Further, we show that when augmented with our personalized retriever, opinion aggregation approaches such as summarization models could seamlessly integrate into the retriever-reader framework. We demonstrate this by proposing two *novel* baselines, where summarization models are used to aggregate retrieved reviews.

PRAG-Optimus (Optimus) Adopts a state-of-the-art pre-trained VAE-based language model based on BERT and GPT (Li et al. 2020). Recent studies find VAE-based language models could be used for unsupervised opinion summarization (Iso et al. 2021). However, Iso et al.'s approach requires searching over large amounts of potential sequences at inference time, particularly when there are multiple inputs. Thus, we use the original Optimus model without inference-time searching as our baseline. Specifically, we fine-tune a pre-trained Optimus model on each dataset for 1 epoch on language modeling, per recommendation by the original authors, and condition the GPT-based generator on the mean embedding of retrieved reviews at inference time.

PRAG-SUM (SUM) Following previous success in training summarizers by learning to recover the target review from a set of distantly retrieved similar reviews (Amplayo and Lapata 2020), we train our summarizer in a leave-one-out fashion over sets of similar reviews. Specifically, for each review in the training set, we retrieve a set of most similar reviews using cosine distance in sentence encoder Φ's semantic space. These retrieved reviews then serve as the input for the model. Then, the summarization module is trained to recover the original review that is used for retrieval from the retrieved reviews. For training, we use the same pre-trained T5 model as PRAG as our base model for a fair comparison and train the model using maximum likelihood estimation (MLE) objective. We provide more details for data processing and training of SUM in the appendix.

6.3 Automatic Evaluation

For automatic evaluation, we generate 10,000 samples from each baseline model, and measure the performance in terms of factuality, informativeness, and generation quality. We provide additional recommendation performance analysis on all eligible models on the whole test set.

Factuality. Factuality constraints that the models' generated explanations are factually correct. Intuitively, the statement in a generated explanation is correct if it could be supported by any existing reviews in the training data. Otherwise, the model is exhibiting **hallucinations** (Maynez et al. 2020). As reported by the same work, entailment models could better measure the factual consistency between reference text and generated content. We follow their findings and evaluate the entailment relationship between generated reviews with reviews of the same product in the training set. Specifically, for each generated explanation, we check

[2]https://github.com/allenai/unifiedqa

[3]https://www.yelp.com/dataset/challenge

[4]https://www.tripadvisor.com

	Entail			D-1			D-2			ENTR			USR			MAUVE		
Method	**Movie**	**Trip**	**Yelp**	**Movie**	**Trip**	**Yelp**	**Movie**	**Trip**	**Yelp**	**Movie**	**Trip**	**Yelp**	**Movie**	**Trip**	**Yelp**	**Movie**	**Trip**	**Yelp**
Att2Seq	25.6	12.2	35.9	39.9	34.6	43.1	75.9	75.4	78.1	9.56	8.11	8.44	41.7	21.0	39.9	3.0	1.4	3.9
NRT	36.1	10.0	31.4	44.0	32.4	41.0	77.8	72.8	76.6	7.5	7.5	8.3	36.1	46.3	44.4	3.0	3.0	4.2
PETER	29.0	17.5	44.5	27.7	26.8	29.5	58.6	60.7	60.4	10.5	10.1	10.7	60.7	57.2	58.2	3.7	2.3	2.2
PEPLER	17.9	11.0	16.0	23.2	23	25.5	51.5	52.2	52.5	11.1	10.0	11.0	52.6	41.7	49.1	1.1	0.4	0.4
Optimus	25.1	22.8	11.5	31.9	32.8	33.2	77.3	77	79.3	10.3	8.5	10.7	**98.5**	92.1	**96.1**	3.5	3.3	4.5
SUM	49	29.5	30.8	22.1	18.7	20	67.1	61	63.7	11.2	10.4	11.5	95.3	**94.7**	94.8	5.8	4.7	5.4
PETER+	40.0	32.6	59.4	43.9	42.6	47.0	78.4	81.9	83.1	9.48	8.53	9.85	60.6	31.5	52.8	12.9	5.3	10.4
PRAG	**88.8**	**80.1**	**86.2**	**45.6**	**39.9**	**47.1**	**84.3**	**82.2**	**84.7**	**12.0**	**12.0**	**11.9**	71.8	76.5	70.4	**23.1**	**42.8**	**20.3**

Table 1: **Automatic evaluation results** on test sets. **Entail** denotes the percentage of entailment among the generated reviews. **D-1** and **D-2** scores denote Distinct-1 and Distinct-2 scores. **ENTR** denotes the geometric mean of n-gram entropy among the generated text. **USR** denotes the ratio of unique sentences generated. **MAUVE** is a distribution based-metric to measure the quality of the generated text. See Section 6.3 for details.

whether the explanation entails *any* reviews from the training set for the same product using a pre-trained entailment model [5]. We note that a generated explanation is factual if it entails any piece of existing reviews for the product, and report the entailment ratio. Specifically, we report the percentage of entailment as Entail.

Informativeness. We measure the informativeness of generated explanations using token, sentence, and corpus level evaluations. Concretely, We evaluate the models using Distinct-1 and Distinct-2 (D-1, D-2) scores (Li et al. 2016), Unique Sentence Ratio (USR) as proposed by (Li, Zhang, and Chen 2020) and ENTR (Jhamtani et al. 2018) following previous work on diversifying generated content (Majumder et al. 2021).

Generation quality. To measure the generated explanation quality, we opt to measure how human-like the generated explanations are. Specifically, we adopt MAUVE (Pillutla et al. 2021), a distribution-based evaluation metric that measures how close the generated contents are to the ground truth corpus.

Recommendation performance. We evaluate the recommendation performance using Rooted Mean Squared Error (RMSE) score following previous works done using the same dataset.

7 Results

7.1 Quantitative Evaluation

We report PRAG's performance compared to baseline models as in Table 1. PRAG consistently outperforms the baseline models in terms of both diversity and informativeness, and can generate high-quality sentences compared to human-written reviews.

Retrieval component improves generation quality. As shown in Table 1, conditioning on additional information improves both model diversity and factuality. Specifically, Optimus, SUM, and PRAG consistently outperform other models that only maintain vector representation for each user and item. The generated content has both higher sentence-wise diversity (from D-1 and D-2 scores) and higher corpus-level diversity (from ENTR and USR scores), as well as being closer to human written reviews (from MAUVE scores). This shows that the personalized retrieval component could improve generation quality.

Training data affects generator factuality. While retrieved historical reviews encourage more factual output in general, this does not guarantee strong factuality. In particular, the T5-based summarizer and Optimus model have limited improvement in terms of factuality compared to other baseline models. This highlights the major cause of hallucination is directly training generators on noisy data. Specifically, Maynez et al. reported that ground-truth sequences in the training data that contain hallucinated content would trigger the model to be less factual at inference time in summarization. In parallel, Longpre et al. reported that noisy retrieved content would cause the generator to hallucinate for question answering.

We note that a similar case applies to review generation as well. In particular, there is no guarantee that the ground truth review will entail *any* of other reviews for the same product in the training set. While this behavior is natural for human (Maynez et al. 2020), the model ended up learning to hallucinate during optimization as a result. By leveraging a reader model trained to faithfully present content in the input text, PRAG sidesteps this issue, and thus has the best factuality performance across models.

7.2 Recommendation Performance

To verify PRAG's ability for recommendation, we report RMSE scores on the three datasets as shown in Table 2. Since SUM and Optimus are based on the same retriever model as PRAG, we report PRAG's performance only in this section. Further, Att2Seq cannot produce rating estimations, and thus we omit the model in the table. We validate that the performance of baseline models is consistent with previous works on the same dataset (Geng et al. 2022). As shown in Table 2, PRAG achieved state-of-the-art rating estimation performance.

[5] https://huggingface.co/prajjwal1/roberta-large-mnli

	Movie	TripAdivor	Yelp
NRT	0.79	0.95	1.01
PETER	0.80	0.95	1.01
PEPLER	1.71	1.25	1.69
PRAG$_u$	0.80	0.95	1.01
PRAG$_{T5}$	0.80	0.96	1.00
PRAG	0.79	0.95	1.01

Table 2: **RMSE scores for recommendation performance**. By default, PRAG uses MPNET sentence embedding and ties $Q_{u,i}$ to γ_i. PRAG $_{T5}$ uses embedding from T5 model and PRAG(u) maps $Q_{u,i}$ to γ_u.

	Fluency			Informativeness		
PRAG vs.	**Movie**	**Trip**	**Yelp**	**Movie**	**Trip**	**Yelp**
NRT	**37**	**16**	**13**	**28**	**1**	**3**
PETER	**20**	**26**	**15**	-2	**12**	**27**
SUM	**8**	**5**	**3**	0	-6	**5**

Table 3: **Human evaluation results** of PRAG versus baseline models. The number reported are the number of *additional* times PRAG win against the opponent model. Note both PRAG and SUM are retriever augmented generators.

7.3 Human Evaluation

We perform human evaluation using 150 test samples with several of the strongest baselines. Specifically, we compare PRAG's performance against NRT, PETER and SUM. We omit Att2Seq and PEPLER since the models are not as performant as other models in recommendation performance, and leave PETER+ out since ground-truth aspects are not always available. Finally, we pick the summarizer as a stronger retriever-augmented baseline (compared to Optimus) based on automatic evaluation results. We compare the generated reviews in terms of (1) **fluency** compared to other models (2) **informativeness** to the user. We report the results in Table 3. Similar to automatic evaluation results, the retriever could reliably boost generation performance: PRAG is consistently ranked as the most fluent model by human evaluators, while SUM has almost on-par performance due to having access to the personalized retriever. Meanwhile, PRAG and SUM almost invariably win in terms of informativeness compared to previous works.

8 Analysis and Discussion

Analysis of marginalization and retrieval reliability. To validate whether marginalizing the predicted embedding before performing retrieval truly uncovers user or item characteristics instead of resulting in random vectors, we perform a simple test by comparing the retrieval result between a pair of mirroring retrievers. Specifically, we hypothesize that given a user and an item, a retriever that ties $Q_{u,i}$ to γ_u and a retriever that ties $Q_{u,i}$ to γ_i should achieve agreement after marginalization. That is, the two retrievers should be able to retrieve similar reviews. To verify this, we report the average exact-match between two retrievers using agreement at 5. As shown in Table 5, the retrievers consistently achieved significantly higher agreement compared to a random baseline across three datasets. Further, we report examples of retrieved reviews with and without marginalization. As shown in Table 4, after marginalization, there is a clear trend that reviews related to aesthetic aspects are being retrieved, as opposed to a set of generic reviews without marginalization.

Retrieved Reviews
the decor of the hotel is greatly refined. ... the building and decoration are very nice and very tastefully decorated. a four seasons in every respect it is an architecturally interesting and esthetically pleasing property in a great location. the rooms are stunning.

Table 4: **Retrieved reviews** (cropped, from top-5 results). With marginalization, the retrieved reviews consistently focus a specific aspect. See appendix for comparison with non-marginalized retrieval results.

	Movie	TripAdivor	Yelp
Retriever	2.87	2.44	2.50
Random	0.54	0.61	0.27

Table 5: **Average agreement-at-5** of mirroring retrievers.

Analysis of sentence embedding model. We hypothesize that any strong sentence embedding model could be used for PRAG. To validate such a hypothesis, we train PRAG model on all three datasets using a pre-trained T5-based sentence encoder. We report the performance as in Table 2. As shown, there is no significant performance variation across different types of sentence embeddings.

Effect of tying user or item Factors to latent query. Similar to the HFT (McAuley and Leskovec 2013) model, the rating estimation component in PRAG could either tie the user or item factor to $Q_{u,i}$. We conduct experiments using both types of architectures, and also report our results in Table 2. Similar to findings reported for the HFT model (McAuley and Leskovec 2013), the performance is generally dataset-dependent, and the design could be viewed as a hyper-parameter.

9 Summary and Outlook

In this work, we propose PRAG, a retriever-reader model that can generate factual and diverse explanations for recommendation. Experiments on three real-world datasets show PRAG can generate both factually grounded and informative explanations. We also investigate the cause of hallucinated content in review generation, and demonstrate the benefit of training text generation models on hallucination-free tasks and datasets. Meanwhile, although we adopted a question-answering model for explanation generation, PRAG's retrieval component could provide support for personalizing a wider range of knowledge-based tasks, such as personalized conversational recommendation, summarization, and product description generation.

References

Amplayo, R. K.; and Lapata, M. 2020. Unsupervised Opinion Summarization with Noising and Denoising. In *ACL*, 1934–1945. Online: Association for Computational Linguistics.

Bražinskas, A.; Lapata, M.; and Titov, I. 2020. Unsupervised Opinion Summarization as Copycat-Review Generation. In *ACL*, 5151–5169. Online: Association for Computational Linguistics.

Bražinskas, A.; Lapata, M.; and Titov, I. 2021. Learning Opinion Summarizers by Selecting Informative Reviews. In *EMNLP*, 9424–9442. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T. J.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. *ArXiv*, abs/2005.14165.

Cheng, H.-T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispir, M.; Anil, R.; Haque, Z.; Hong, L.; Jain, V.; Liu, X.; and Shah, H. 2016. Wide & Deep Learning for Recommender Systems. In *1st Workshop on Deep Learning for Recommender Systems*, DLRS 2016, 7–10. New York, NY, USA: Association for Computing Machinery. ISBN 9781450347952.

Dalvi, B.; Jansen, P.; Tafjord, O.; Xie, Z.; Smith, H.; Pipatanangkura, L.; and Clark, P. 2021. Explaining Answers with Entailment Trees. In *EMNLP*, 7358–7370. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Dong, L.; Huang, S.; Wei, F.; Lapata, M.; Zhou, M.; and Xu, K. 2017. Learning to Generate Product Reviews from Attributes. In *EACL*, 623–632. Valencia, Spain: Association for Computational Linguistics.

Geng, S.; Fu, Z.; Ge, Y.; Li, L.; de Melo, G.; and Zhang, Y. 2022. Improving Personalized Explanation Generation through Visualization. In *ACL*, 244–255. Dublin, Ireland: Association for Computational Linguistics.

Guu, K.; Lee, K.; Tung, Z.; Pasupat, P.; and Chang, M.-W. 2020. REALM: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

Hada, D. V.; Vijaikumar, M.; and Shevade, S. K. 2021. ReXPlug: Explainable Recommendation using Plug-and-Play Language Model. *SIGIR*.

Hancock, B.; Varma, P.; Wang, S.; Bringmann, M.; Liang, P.; and Ré, C. 2018. Training Classifiers with Natural Language Explanations. In *ACL*, 1884–1895. Melbourne, Australia: Association for Computational Linguistics.

He, R.; and McAuley, J. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. *WWW*.

Hokamp, C.; and Liu, Q. 2017. Lexically Constrained Decoding for Sequence Generation Using Grid Beam Search. In *ACL*, 1535–1546. Vancouver, Canada: Association for Computational Linguistics.

Iso, H.; Wang, X.; Suhara, Y.; Angelidis, S.; and Tan, W.-C. 2021. Convex Aggregation for Opinion Summarization. In *Findings of EMNLP*, 3885–3903. Punta Cana, Dominican Republic: Association for Computational Linguistics.

Jhamtani, H.; Gangal, V.; Hovy, E.; Neubig, G.; and Berg-Kirkpatrick, T. 2018. Learning to Generate Move-by-Move Commentary for Chess Games from Large-Scale Social Forum Data. In *ACL*, 1661–1671. Melbourne, Australia: Association for Computational Linguistics.

Lewis, P. S. H.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.; Rocktäschel, T.; Riedel, S.; and Kiela, D. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *NeurIPS*.

Li, C.; Gao, X.; Li, Y.; Peng, B.; Li, X.; Zhang, Y.; and Gao, J. 2020. Optimus: Organizing Sentences via Pre-trained Modeling of a Latent Space. In *EMNLP*, 4678–4699. Online: Association for Computational Linguistics.

Li, J.; Galley, M.; Brockett, C.; Gao, J.; and Dolan, B. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In *NAACL-HLT*, 110–119. San Diego, California: Association for Computational Linguistics.

Li, L.; Zhang, Y.; and Chen, L. 2020. Generate Neural Template Explanations for Recommendation. In *CIKM*, CIKM '20, 755–764. New York, NY, USA: Association for Computing Machinery. ISBN 9781450368599.

Li, L.; Zhang, Y.; and Chen, L. 2021. Personalized Transformer for Explainable Recommendation. In *ACL-IJCNLP*, 4947–4957. Online: Association for Computational Linguistics.

Li, L.; Zhang, Y.; and Chen, L. 2022. Personalized Prompt Learning for Explainable Recommendation. *arXiv preprint arXiv:2202.07371*.

Li, P.; Wang, Z.; Ren, Z.; Bing, L.; and Lam, W. 2017. Neural Rating Regression with Abstractive Tips Generation for Recommendation. *SIGIR*.

Liu, S.; Sabour, S.; Zheng, Y.; Ke, P.; Zhu, X.; and Huang, M. 2022. Rethinking and Refining the Distinct Metric. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *ACL*, 762–770. Association for Computational Linguistics.

Longpre, S.; Perisetla, K.; Chen, A.; Ramesh, N.; DuBois, C.; and Singh, S. 2021. Entity-Based Knowledge Conflicts in Question Answering. In *EMNLP*, 7052–7063. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Majumder, B. P.; Berg-Kirkpatrick, T.; McAuley, J.; and Jhamtani, H. 2021. Unsupervised Enrichment of Persona-grounded Dialog with Background Stories. In *ACL-IJCNLP*,

585–592. Online: Association for Computational Linguistics.

Majumder, B. P.; Camburu, O.; Lukasiewicz, T.; and McAuley, J. J. 2022. Rationale-Inspired Natural Language Explanations with Commonsense. *ICML*, abs/2106.13876.

Marasović, A.; Bhagavatula, C.; Park, J. s.; Le Bras, R.; Smith, N. A.; and Choi, Y. 2020. Natural Language Rationales with Full-Stack Visual Reasoning: From Pixels to Semantic Frames to Commonsense Graphs. In *Findings of EMNLP*, 2810–2829. Online: Association for Computational Linguistics.

Maynez, J.; Narayan, S.; Bohnet, B.; and McDonald, R. 2020. On Faithfulness and Factuality in Abstractive Summarization. In *ACL*, 1906–1919. Online: Association for Computational Linguistics.

McAuley, J.; and Leskovec, J. 2013. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. In *ACM RecSys*, RecSys '13, 165–172. New York, NY, USA: Association for Computing Machinery. ISBN 9781450324090.

Ni, J.; Lipton, Z. C.; Vikram, S.; and McAuley, J. 2017. Estimating Reactions and Recommending Products with Generative Models of Reviews. In *IJCNLP*, 783–791. Taipei, Taiwan: Asian Federation of Natural Language Processing.

Ni, J.; and McAuley, J. 2018. Personalized Review Generation By Expanding Phrases and Attending on Aspect-Aware Representations. In *ACL*, 706–711. Melbourne, Australia: Association for Computational Linguistics.

Pagnoni, A.; Balachandran, V.; and Tsvetkov, Y. 2021. Understanding Factuality in Abstractive Summarization with FRANK: A Benchmark for Factuality Metrics. In *NAACL-HLT*, 4812–4829. Online: Association for Computational Linguistics.

Pillutla, K.; Swayamdipta, S.; Zellers, R.; Thickstun, J.; Welleck, S.; Choi, Y.; and Harchaoui, Z. 2021. MAUVE: Measuring the Gap Between Neural Text and Human Text using Divergence Frontiers. In *NeurIPS*.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.*, 21: 140:1–140:67.

Ram, O.; Kirstain, Y.; Berant, J.; Globerson, A.; and Levy, O. 2021. Few-Shot Question Answering by Pretraining Span Selection. In *ACL-IJCNLP*, 3066–3079. Online: Association for Computational Linguistics.

Song, K.; Tan, X.; Qin, T.; Lu, J.; and Liu, T. 2020. MPNet: Masked and Permuted Pre-training for Language Understanding. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *NeurIPS*.

Tafjord, O.; and Clark, P. 2021. General-Purpose Question-Answering with Macaw. *CoRR*, abs/2109.02593.

Wiegreffe, S.; Marasović, A.; and Smith, N. A. 2021. Measuring Association Between Labels and Free-Text Rationales. In *EMNLP*, 10266–10284. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Davison, J.; Shleifer, S.; von Platen, P.; Ma, C.; Jernite, Y.; Plu, J.; Xu, C.; Le Scao, T.; Gugger, S.; Drame, M.; Lhoest, Q.; and Rush, A. 2020. Transformers: State-of-the-Art Natural Language Processing. In *EMNLP: System Demonstrations*, 38–45. Online: Association for Computational Linguistics.

Zhang, Y.; Lai, G.; Zhang, M.; Zhang, Y.; Liu, Y.; and Ma, S. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In Geva, S.; Trotman, A.; Bruza, P.; Clarke, C. L. A.; and Järvelin, K., eds., *SIGIR*, 83–92. ACM.

Zhou, G.; Zhu, X.; Song, C.; Fan, Y.; Zhu, H.; Ma, X.; Yan, Y.; Jin, J.; Li, H.; and Gai, K. 2018. Deep Interest Network for Click-Through Rate Prediction. In *SIGKDD*, KDD '18, 1059–1068. New York, NY, USA: Association for Computing Machinery. ISBN 9781450355520.

Zhou, W.; Hu, J.; Zhang, H.; Liang, X.; Sun, M.; Xiong, C.; and Tang, J. 2020. Towards Interpretable Natural Language Understanding with Explanations as Latent Variables. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *NeurIPS*.

Zhou, W.; Li, Q.; and Li, C. 2021. Learning from Perturbations: Diverse and Informative Dialogue Generation with Inverse Adversarial Training. In *ACL-IJCNLP*, 694–703. Online: Association for Computational Linguistics.