12-2022

Question-attentive review-level recommendation explanation

Trung Hoang LE
Singapore Management University, thle.2017@phdcs.smu.edu.sg

Hady Wirawan LAUW
Singapore Management University, hadywlauw@smu.edu.sg

Question-Attentive Review-Level Recommendation Explanation

Trung-Hoang Le
School of Computing and Information Systems
Singapore Management University
Singapore
thle.2017@smu.edu.sg

Hady W. Lauw
School of Computing and Information Systems
Singapore Management University
Singapore
hadywlauw@smu.edu.sg

***Abstract*—Recommendation explanations help to improve their acceptance by end users. The form of explanation of interest here is presenting an existing review of the recommended item. The challenge is in selecting a suitable review, which is customarily addressed by assessing the relative importance of each review to the recommendation objective. Our focus is on improving review-level explanation by leveraging additional information in the form of questions and answers (QA). The proposed framework employs QA in an attention mechanism that aligns reviews to various QAs of an item and assesses their contribution jointly to the recommendation objective. The benefits are two-fold. For one, QA aids in selecting more useful reviews. For another, QA itself could accompany a well-aligned review in an expanded form of explanation. Experiments showcase the efficacies of our method as compared to baselines in identifying useful reviews and QAs, while maintaining parity in recommendation performance.**

I. Introduction

Earlier in the evolution of recommender systems, the concern was predominantly on achieving higher accuracies [1]–[3]. Of late, the concern shifts to greater interpretability and explainability, as ultimately the goal is to get users to adopt the recommendations. This gives rise to a plethora of explainable recommendation models [4], which seek to produce not only recommendations, but also accompanying explanations.

For a pertinent instance, we allude to *review-level explanation*, whereby the explanation to a recommendation takes the form of a review, selected from the existing reviews of the product. For instance, on Amazon.com, Canon EOS Rebel T7 Bundle[1] has more than 2800 ratings, more than 300 of which have reviews. One of these reviews is illustrated in Figure 1, relating the quality of the starter kit. With a rich corpus of reviews comes the problem in how to select which review to present as an explanation. One paradigm [5], [6] is to weigh the contribution of reviews to the recommendation objective. An insightful review, when presented with a recommended product, allows the recipient to empathize with the hands-on experience of the reviewer, thus anticipating what her own experience with the product would be.

In this work, we go beyond reviews and incorporate other information such as a question posted by a user that attracts answers from other users, referred to in short form as QA. For

[1]https://www.amazon.com/Canon-T7-18-55mm-3-5-5-6-Accessory/dp/B07P15K8Q7/

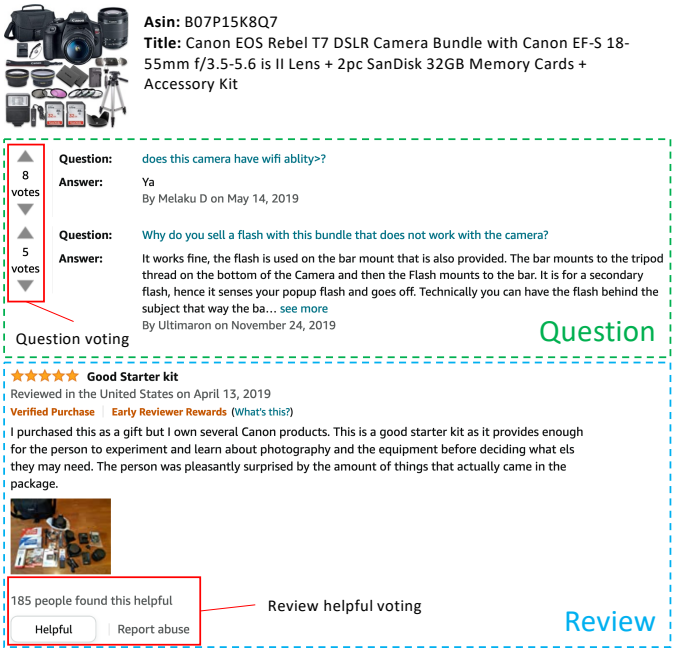

Fig. 1. A product with question and review

instance, the product in Figure 1 has more than 200 questions, including whether the camera has wifi ability (*answer: yes*), whether there is a port for an external microphone (*answer: no, but another model T7i does*), and whether it is suitable for indoor sports (*answer: yes, it has a sport mode*).

Interestingly, questions and their answers present a distinct yet complementary information to reviews. Where reviews tend to be subjective and replete with opinions, questions tend to be objective and inquisitive of factual concerns. Where a single review tends to be multi-faceted and comprehensive, each question tends to be concise and narrowly focused on a single aspect. Given this complementarity, we postulate that both QA and review could collectively serve as recommendation explanations. The former notifies the recommendee of relevant factual concern(s), while the latter gains the recommendee insights from a reviewer's experience.

Problem. Let $\mathcal{U}$ be a set of users, and $\mathcal{P}$ be a set of products. A user $i \in \mathcal{U}$ assigns to a product $j \in \mathcal{P}$ a rating

$r_{ij} \in \mathbb{R}_+$ along with a review t_{ij}. We denote the collection of all ratings as $\mathcal{R}$, that of all reviews as $\mathcal{T}$, and the subset of reviews concerning a product j as $\mathcal{T}_j$. Product j may also have multiple questions $\mathcal{Q}_j = \{q_{j1}, q_{j2}, ..., q_{j|Q_j|}\} \subset \mathcal{Q}$. Each question is presumed to be accompanied by answer(s), collectively referred to in short form as QA.

The problem can thus be stated as follows. Receiving as input users $\mathcal{U}$, products $\mathcal{P}$, ratings $\mathcal{R}$, reviews $\mathcal{T}$, and question-answer pairs $\mathcal{Q}$, we seek a model capable of predicting a missing rating by a user i on product j for recommendation (rating regression), as well as identifying a QA pair (selected from $\mathcal{Q}_j$) along with a review (selected from $\mathcal{T}_j$) to serve collectively as explanations accompanying the recommendation.

Due to the differing yet complementary natures of QA and reviews, we design a neural attention model, called QUESTER, that operates at two levels. First, the concise QA serves as focal points of attention representing salient aspects to a product recommendation. Second, the multi-faceted nature of reviews means that they could be relevant to multiple aspects, and we model their relative importance to each QA. Together, QA and reviews serve dual roles in a hand-in-hand manner: to contribute content features to aid recommendation and to serve as explanations to a recommendation.

Contributions. *First*, to our best knowledge, this is the first work to incorporate product questions into an attention mechanism on reviews for recommendation. *Second*, we develop a neural model called QUESTion-attentive review-level Explanation for neural rating Regression or QUESTER, which considers questions as a source of alignment to textual review. *Third*, we conduct comprehensive experiments against baselines that showcase the effectiveness of our approach.

II. Related Work

Our work belongs to a category of recommender systems that use whole reviews as explanations. NARRE [5] uses attention to weigh each individual review toward user and item representation and uses the most useful review(s) as review-level explanation. HRDR [6] uses multilayer perceptron to encode user's ratings (resp. item's ratings) as user features (resp. item's features) and use that as query for attention layer to weight the contribution of each review to rating prediction. HFT [7] could select the review with the closest topic distribution to the item's topic distribution. Our key distinction from these baselines is our unique incorporation of QA both for review selection and explanation. The use of QA in for recommendation is still rare. One that is distinct from our scenario is detecting a user's propensity to purchase a product based on the question the user has submitted [8].

III. Methodology

We hypothesize that the concise questions could serve as an attention mechanism in weighting the importance of reviews. The overall architecture of our proposed QUESTER model is shown in Figure 2. Below we describe its various components.

Text Encoder. We use a widely adopted CNN text processor [5], [6], [9], named TEXTCNN, for encoding to extract

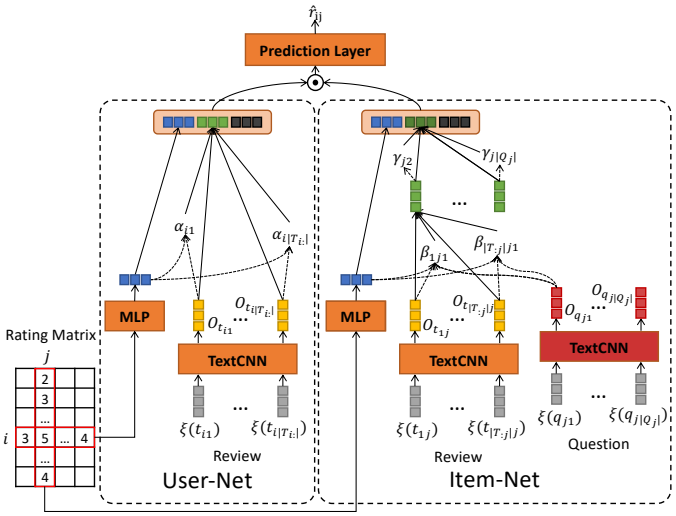

Fig. 2. QUESTER model

semantic features of text. TEXTCNN consists of a Convolutional Neural Network (CNN) followed by max pooling and a fully connected layer. Particularly, we have a word embedding function $\xi : M \rightarrow \mathbb{R}^D$ to map each word in the text t into a D-dimensional vector, forming an embedded matrix $\xi(t)$ with fixed length W (padded zero for text with length $< W$). Following this embedding layer is a convolutional layer with m neurons, each associated with a filter $F \in \mathbb{R}^{w \times D}$, each k^{th} neuron produces features by applying convolution operator on the embedded matrix $\xi(t)$:

$$z_k = ReLU(\xi(t) * F_k + b_z) \tag{1}$$

$ReLU(x) = \max(x, 0)$ is a nonlinear activation function and $*$ is the convolution operation. With sliding window w, the produced features would be $z_1, z_2, ..., z_k^{W-w+1}$, which are passed to a max pooling to capture the most important features having highest values, which is defined as:

$$o_k = \max(z_1, z_2, ..., z_k^{W-w+1}) \tag{2}$$

We get the final output of the convolutional layer by concatenating all output from m neurons, $O = [o_1, o_2, ..., o_m]$. A simple approach to get the final representation of the input text t is to pass O into a fully connected layer as follows:

$$X = WO + b \tag{3}$$

Rating Encoder. Ratings are explicit features provided by users to indicate their interest on given items. The user ratings $r_{i:}$ form a rating pattern for user i, and the item ratings $r_{:j}$ form a rating pattern for item j. A reasonable choice is to use a multi-layer perceptron (MLP) network to learn the representation for the rating pattern [6]. Specifically,

$$\begin{aligned} h_{i1} &= \tanh(W_{r_{i:}1} r_{i:} + b_{r_{i:}1}) \\ h_{i2} &= \tanh(W_{r_{i:}2} h_{i1} + b_{r_{i:}2}) \\ &... \\ u_i &= \tanh(W_{r_{i:}k} h_{i(k-1)} + b_{r_{i:}k}) \end{aligned} \tag{4}$$

The output u_i is the final rating-based representation of user i, h_{ik} is the output hidden representation at layer k of the MLP. Similarly, we get the rating-based representation p_j of product j from its input ratings $r_{:j}$ in similar manner. We use $\tanh$ as activation function to project the learned rating-based representation into the same range of text-based representations that will be discussed in the following paragraphs.

User Attention-Based Review Pooling. Equation 3 presumes that the contribution of each review is the same towards the final representation. The importance of each review contributing to user final representation is learnt as follows:

$$\rho_{ij} = \tanh(W_{O_t}(O_{t_{ij}} \odot u_i) + b_\rho) \tag{5a}$$

$$\theta_{ij} = W_\rho \rho_{ij} + b_\theta \tag{5b}$$

$$\alpha_{ij} = \frac{\exp(\theta_{ij})}{\sum_i \exp(\theta_{ij})} \tag{5c}$$

where $\odot$ is element-wise multiplication operator, u_i is the rating-based representation of the user i, $O_{t_{ij}}$ is the feature vector extracted from review text t_{ij} by TEXTCNN, α_{ij} is the normalized attention score of the review t_{ij}, which can be interpreted as the contribution of that review to the feature profile O_i of user i, aggregating as follows:

$$O_i = \sum_j \alpha_{ij} O_{t_{ij}} \tag{6}$$

The final representation of user i is computed as follows:

$$X_i = W_{O_i} O_i + b_X \tag{7}$$

Item Question-Attentive Review-Level Explanations. A naive approach to model question on item side is to apply similar approach of modeling reviews. However, the connection between reviews and questions would have been overlooked. Here we presume that a review may contain information that could be relevant to a question. We aggregate another attention layer based on item questions that help us to incorporate reviews based on their contribution towards item questions.

In particular, let $O_{t_{ij}}$ be the review encoding and $O_{q_{jk}}$ be the QA encoding of the product j. With respect to each question representation $O_{q_{jk}}$, we learn the attention weights β_{ijk} for review representation $O_{t_{ij}}$ by projecting both question and review representation onto an attention space followed by a non-linear activation function; the outputs are ϕ_{jk} and ρ'_{ij} respectively. We use $\tanh$ activation function to scale $O_{q_{jk}}$ and $O_{t_{ij}}$ to the same range of values, so that neither component dominates the other. To learn the question-specific attention weight of a review, we let the question projection ϕ_{jk} interact with the review projection ρ'_{ij} in two ways: element-wise multiplication and summation. The learned vector V plays the role of global attention context. This produces an attention value η_{ijk}, which is normalized using softmax to obtain β_{ijk}:

$$\phi_{jk} = \tanh(W_{O_q} O_{q_{jk}} + b_\phi) \tag{8a}$$

$$\rho'_{ij} = \tanh(W_{O_t}(O_{t_{ij}} \odot p_j) + b_{\rho'}) \tag{8b}$$

$$\eta_{ijk} = V^T(\phi_{jk} \odot \rho'_{ij} + \rho'_{ij}) \tag{8c}$$

$$\beta_{ijk} = \frac{\exp(\eta_{ijk})}{\sum_i \exp(\eta_{ijk})} \tag{8d}$$

Using the question-specific attention weights β_{ijk}, we aggregate the review representations $O_{t_{ij}}$'s into a question-specific representation d_{jk} as follows.

$$d_{jk} = \sum_i \beta_{ijk} O_{t_{ij}} \tag{9}$$

For a document (a product question with all of its reviews), we apply this attention mechanism for every product question, yielding a set of question-specific document representations d_{jk}, $k \in [1, |Q_j|]$. All the d_{jk}'s need to be aggregated into the final document representation O_j before incorporating to product representation. Thus, we seek to learn the importance weight γ_{jk}, signifying how each question-specific representation d_{jk} would contribute to O_j.

$$\kappa_{jk} = K^T \tanh(W_{d_{jk}} d_{jk} + b_\kappa) \tag{10a}$$

$$\gamma_{jk} = \frac{\exp(\kappa_{jk})}{\sum_k \exp(\kappa_{jk})} \tag{10b}$$

Question-specific representation d_{jk} is projected into attention space through a layer of neurons with non-linear activation function $\tanh$. The scalar κ_{jk} indicates the importance of d_{jk}, obtained by multiplying with global attention context vector K (randomly initialized and learned during training). The representation d_{jk}'s due to the various questions are aggregated into the final product representation O_j using soft attention pooling with attention weight γ_{jk}'s.

$$O_j = \sum_k \gamma_{jk} d_{jk} \tag{11a}$$

$$Y_j = W_{O_j} O_j + b_Y \tag{11b}$$

Prediction Layer. The latent factors of user i and product j are mapped to a shared hidden space as follows:

$$h_{ij} = [u_i; X_i; \zeta_u(i)] \odot [p_j; Y_j; \zeta_p(j)] \tag{12}$$

where $\zeta_u(\cdot)$ and $\zeta_p(\cdot)$ are embedding function to map each user and each product into their embedding space respectively, X_i is user preferences and Y_j is item features obtained from user reviews and product reviews and questions, $[u_i; X_i; \zeta_u(i)]$ is the concatenation of user rating-based representation u_i, user text attention review pooling X_i, and user i embedding $\zeta_u(i)$. The final rating prediction is computed as follows:

$$\hat{r}_{ij} = W^T h_{ij} + b_i + b_j + \mu \tag{13}$$

Learning. Similar to prior works on rating prediction task [5], [6], [10], which is a regression problem, we adopt the squared loss function:

$$\mathcal{L} = \sum_{i,j \in \Omega} (\hat{r}_{ij} - r_{ij})^2 \tag{14}$$

Where Ω denotes the set of all training instances, r_{ij} is the ground truth rating that user i assigned on product j.

The most important question L is selected by $L = argmax_k(\gamma_{jk})$ and the most useful review is selected by $argmax_i(\beta_{ijL})$. We use the selected question with its answer and the selected review collectively as explanation.

A limitation of relying only on questions found within a product is that product features may not be captured completely, because some products do not have sufficient questions to cover all its important aspects. As a result, an important review may be overlooked because it does not correspond to any question. To address this limitation, in addition to the questions found in a product, we include one more global "General Question", which allows those important reviews to still be aligned. This additional question plays the role of "global" aspect, and also helps our model to potentially generalize to product without questions.

IV. Experiments

As this work is primarily about recommendation explanations, rather than rating prediction per se, and the two objectives are not necessarily directionally equivalent, our orientation is to improve explanations while maintaining parity in accuracy performance. In particular, our core contribution is in incorporating question and answer or QA for review-level explanation. The experimental objectives revolve around the utility of QA as part of explanation, the effectiveness of QA to aid the selection of review-level explanation, and the alignment of QA and review that are part of an explanation.

Datasets. Towards reproducibility, we work with publicly available sources. While QA is a feature on many platforms, not many such datasets have both reviews and QA information. One that does is the Amazon Product Review Dataset[2] [11]. We experiment on three product categories from this source as separate instances. These categories are selected for significant availability of QA information. Consistent performance across multiple categories with different statistics bolster the analysis. Table I summarizes basic statistics of the three datasets.

For greater coverage, we collect item questions and acquire their helpfulness voting scores from the Amazon.com website. Too short reviews (less than 3 words), users and items with fewer than five reviews are filtered out. Code and datasets are available at https://github.com/PreferredAI/QuestER. For each question, we also include one answer (the earliest that appears in the data) as frequently answers are similar. To aggregate overlapping questions, we cluster questions with KMeans, keeping questions from big clusters which cover 80% of questions. For smaller clusters, we keep the nearest question to each cluster centroid and combine them into "General Question" (all products have this by default). This is used solely for modeling to generalize to items without questions, but would not be used as a recommendation explanation.

Baselines. We evaluate our proposed QUESTER against the following baselines in terms of useful review and QA selection. Comparisons between methods are tested with one-tailed paired-sample Student's t-test at 0.05 level.

- **HRDR** [6] uses attention mechanism with the rating-based representation as features to weight the contribution of each individual review toward user/item representation.

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

TABLE I
DATA STATISTICS

Dataset	#Item	#User	#Review (Rating)	#Question	$\frac{\#\text{Item with Question}}{\#\text{Item}}$
Home	28,169	66,295	549,895	368,904	0.3193
Sport	18,301	35,447	295,074	123,119	0.1940
Musical	893	1,416	10,163	22,409	0.5622

- **NARRE** [5] learns to predict ratings and the usefulness of each reviews by applying attention mechanism for reviews on users/items embedding.
- **HFT** [7] models the latent factors from user or item reviews by employing topic distributions. In this work, we employ item reviews and applied their proposed usefulness review retrieval approach for selecting useful reviews. The number of topics is $K = 50$.

Note that our key distinction from the above mentioned baselines is that we further incorporate product questions. As there is no prior work on predicting ratings along with selecting useful question, when the evaluative task is to look into selecting questions (question retrieval and question similarity tasks, see Section IV-A and Section IV-C), we would apply similar approach for each baseline such that item text will be item questions instead of item reviews.

Training Details. Each item's reviews are split randomly into train, validation, and test with ratio $0.8 : 0.1 : 0.1$. Unknown users are excluded from validation and test sets. We employ the pretrained word embeddings from GloVe [12] to initialize the text embedding matrix with dimensionality of 100 in which the embedding matrix is shared for both reviews and questions. We use separate TEXTCNN for user reviews, item reviews, and item QAs. Max text length W is 128, the number of neurons in convolutional layer m is 64, the window size w is 3. The latent factor number $k \in \{8, 16, 32, 64\}$. After tuning, we set $k = 8$ for memory efficiency as using larger k does not improve the performance significantly. Dropout ratio is 0.5 as in [5]. We apply 3-layers MLP for rating-based representation modeling as in [6], with the number of neural units in hidden layers to be $\{|l|, 128, 64, m\}$ where $|l|$ is the number of items (resp. number of users) for user-net (resp. item-net). Using Adam optimizer [13] with an initial learning rate of 10^{-3} and mini-batch size of 64, we see models tend to converge before 20 epochs. We set a maximum of 20 epochs and report the test result from the best performing model (MSE) on validation, a uniform practice across methods.

Brief Comment on Running Time. Our focus in this work is recommendation explanation, rather than computational efficiency. The models can be run offline. For a sense of the running times, our model takes between 5 minutes on the Musical category to 5 hours on the Home category on AMD EPYC 7742 64-Core Processor and NVIDIA Quadro RTX 8000. The running times of the baselines are generally in the same ballpark.

A. Question and Review Alignment

Our proposed recommendation explanation consists of a question-and-answer (QA) and a review. Ideally, these two

TABLE II
PERFORMANCE IN QUESTION AND REVIEW ALIGNMENT

Data	Model	R-1	R-2	R-L	METEOR
Home	QUESTER	**15.73**§	**0.93**§	**7.91**§	**10.27**§
	HRDR	14.71	0.74	6.91	8.07
	NARRE	14.70	0.72	6.75	7.72
	HFT	13.53	0.65	6.38	7.49
Sport	QUESTER	**15.92**§	**0.80**§	**7.83**§	**10.05**§
	HRDR	14.96	0.60	6.72	7.77
	NARRE	14.15	0.51	5.86	6.51
	HFT	13.86	0.56	6.09	7.27
Musical	QUESTER	**16.57**§	**0.94**§	**7.54**§	**11.64**§
	HRDR	15.15	0.72	6.61	9.66
	NARRE	15.53	0.75	6.86	8.35
	HFT	12.94	0.59	5.88	8.72

§ denotes statistically significant improvements. Highest values are in **bold**.

components, QA on one hand, and review on the other hand, are well-aligned for a more coherent explanation. We measure this alignment using ROUGE [14] and METEOR [15], two well-known metrics for text matching and text summarization. To cater to words as well as phrases, we report F-Measure of ROUGE score measuring the overlapping unigrams (R-1), bigrams (R-2), and the longest common subsequence (R-L) between the reference summary and evaluated summary. We compute ROUGE and METEOR scores for the top-1 selected question and review and report them in Table II.

The results show that the proposed QUESTER consistently outperforms the baselines significantly across virtually all the datasets. This shows QUESTER's QAs and reviews that are part of a collective explanation are better-aligned with each other, as compared to the respective pairings identified by the baselines. Note that HRDR, NARRE, and HFT had been designed solely to select helpful reviews. To be able to compare with these models, we ran each model twice, once with reviews and another time replacing item reviews with QA's. This approach essentially treats review and question in a disjoint manner, which contributes to why they are underperforming as compared to our proposed QUESTER that jointly selects review and question that are well-aligned with each other.

B. Review-Level Explanation

Here we assess whether incorporating questions would help in selecting reviews for the explanation. We take reviews that have the greatest positive helpfulness voting scores on every product to be the ground truth to study the performance of selecting useful reviews. We use Precision at 5 (Prec@5), Recall at 5 (Rec@5), and F1@5 as evaluation. As reported in Table III (left), our proposed QUESTER is the better-performing method overall. Its outperformance over baseline models is statistically significant.

To further assess the quality of top-ranked reviews against top-rated helpful reviews, we again use ROUGE and METEOR as metrics. The results in Table III (left) consistently show that our proposed QUESTER outperforms all baseline models significantly in all measurements, i.e., the top-ranked reviews from QUESTER are more similar to the top-rated helpful reviews than those of HRDR, NARRE, and HFT. Overall, in addition to the reviews, our QUESTER uses additional product QA, achieving better results than the baseline methods those only use reviews as additional data, suggesting that using QA aids in selecting more useful reviews.

C. Question-Level Explanation

The novelty of the proposed QUESTER is in producing question-level explanation along with review-level explanation. We conduct a homologous quantitative evaluation as Review-Level Explanation above, but now with question votes as ground-truth and measure Prec@5, Rec@5, and F1@5. In addition, we measure the similarity between question generated by QUESTER and top voted useful question using ROUGE and METEOR, the first answer of each question is concatenated as a part of the question text for evaluation. As shown in Table III (right), QUESTER is competitive throughout. In many cases, it shows better results than the baselines, and frequently in a statistically significant manner. Notably, a baseline never beats the proposed method in a statistically significant manner.

D. Rating Prediction

As previously established, our main focus in this work is on recommendation explanations, with an eye on improving the selection of reviews and incorporating questions in that endeavour. Nevertheless, while recommendation accuracy is not the main focus, we find that QUESTER still maintains parity in this regard with the other methods.

We report the average of *Mean Square Error* (MSE) averaged across users on each category in Table IV. While the performances of various methods vary slightly across categories, the average MSE across categories (the last row) for QUESTER is slightly lower (better). Our proposed QUESTER achieves comparable results when compared to the neural models HRDR and NARRE.

E. Case Study

To investigate the usefulness of the recommendation explanation consisting of a QA as well as a review, we show a case study that benchmarks QUESTER to the most voted question and the most voted review. Figure 3 shows explanation for a guitar rest. Notably, the pairing by Top_Rated_Useful are not so coherent, with the QA discusses its use for guitars, while the review discusses its use for ukuleles. In contrast, both the QA and the review by QUESTER focus on the key issue of how well the item could hold a guitar in rest. QUESTER's QA is more aligned with its review than those of Top_Rated_Useful, ROUGE-L F-Measures are 14.71 and 6.64 respectively.

V. CONCLUSION

QUESTER is a framework for incorporating question-answer pair or QA into review-based recommendation explanation. We model QA in an attention mechanism to identify more useful reviews. Through joint modeling, we can collectively form an explanation in terms of QA and review.

TABLE III
PERFORMANCE IN REVIEW-LEVEL EXPLANATION AND QUESTION-LEVEL EXPLANATION TASKS

Data	Model	Review-Level Explanation							Question-Level Explanation						
		Prec@5	Rec@5	F1@5	R-1	R-2	R-L	METEOR	Prec@5	Rec@5	F1@5	R-1	R-2	R-L	METEOR
Home	QUESTER	**0.147**§	**0.643**§	**0.234**§	**36.35**§	**20.41**§	**26.56**§	**31.25**§	**0.086**§	**0.325**§	**0.130**§	**23.07**§	**9.36**§	**16.10**§	**19.67**§
	HRDR	0.133	0.574	0.211	30.94	15.16	21.21	24.24	0.082	0.309	0.125	19.70	7.13	12.98	16.13
	NARRE	0.134	0.580	0.213	29.70	13.94	19.98	23.69	0.083	0.309	0.125	19.05	6.40	12.13	15.46
	HFT	0.140	0.611	0.223	28.76	14.21	19.85	23.23	0.082	0.312	0.125	18.40	7.43	13.19	15.00
Sport	QUESTER	**0.159**§	**0.671**§	**0.251**§	**37.24**§	**22.01**§	**27.86**§	**33.50**§	**0.093**	**0.360**	**0.143**	**23.15**§	**9.86**	**16.22**§	**20.87**§
	HRDR	0.146	0.611	0.230	30.87	15.32	21.34	26.15	0.085	0.329	0.131	15.21	3.37	7.94	12.97
	NARRE	0.140	0.583	0.220	26.50	11.44	17.16	20.43	0.088	0.336	0.135	18.31	6.25	11.71	15.28
	HFT	0.155	0.654	0.245	29.80	15.70	21.14	24.92	0.091	0.346	0.139	20.01	9.02	14.91	16.92
Musical	QUESTER	**0.179**§	**0.763**§	**0.284**§	**37.29**	**21.78**	**27.58**	**35.11**	0.082	0.333	0.128	**22.93**	**8.82**	**14.81**	**19.79**
	HRDR	0.173	0.733	0.274	35.81	20.59	26.16	32.84	**0.085**	**0.335**	**0.131**	17.19	3.67	9.35	13.22
	NARRE	0.161	0.677	0.255	27.44	12.08	17.71	21.65	0.078	0.312	0.121	21.90	6.71	13.30	18.25
	HFT	0.173	0.730	0.274	30.86	16.75	21.91	26.66	0.082	0.328	0.127	17.22	5.57	11.35	12.27

§ denotes statistically significant improvements. Highest values are in **bold**.

TABLE IV
RATING PREDICTION PERFORMANCE: MEAN SQUARE ERROR

Data	HFT	NARRE	HRDR	QUESTER
Home	1.2796	1.2654	1.2666	1.2661
Sport	1.0231	1.0054	1.0055	1.0046
Musical	1.0627	0.8889	0.8861	0.8788
Average	1.1218	1.0532	1.0527	1.0498

Asin: B004N0MKN8
Title: Planet Waves Guitar Rest

Top Rated Useful Question: What is the response to the numerous customer reviews that say that the thing keeps falling off unless the guitar is resting against it?
Answer: It does tend to fall off, like you say, but it really is great to lean the guitar on. Otherwise the guitar just falls over. Pick your poison! Sorry
Top Rated Useful Review: The Planet Waves Guitar Rest works for ukuleles! I just got one, and have used it for a few days, and it's the bomb! I can set my little ukuleles down now without fear of falling over. This product is a rubber disc with small "arms" in a gentle curve that nestles against the edge of any surface, and you can set your instrument against it, and voila, it doesn't fall over! Here at home, I use it on the second shelf of a bookcase, and my concert sized ukulele fits like a glove, heel on carpet, neck in Guitar Rest. I'm going to buy a couple more for my ukulele cases, because I can use them at one of my uke parties. If one sets a tiny ukulele on the floor, for instance, to take a whizz, they're just small enough to go unseen and have someone step on them. Here, I just find a spot near wherever I'm sitting, and it becomes my "lean" spot, and I can even set my beer can on the round part on the back! Coaster uke/guitar holder. It's quite immovable once it has some weight against it from the instrument. I could carry a metal stand with me, but it wouldn't fit in my ukulele case--this Planet Waves product does. A winner.

QUESTER Question: Will this guitar rest work on a round table top?
Answer: That depends entirely on the dimensions of the table. Take the guitar and see if it can lay flat across the table. If it does, then it will work just fine. If it goes off the end a little bit, it should still be fine.
QUESTER Review: I've had this thing for several weeks, and just now, when it fell off the table for the 100th time, I tossed it in the trash. The whole thing is one piece of soft floppy rubber, it's not stiff enough for the part that cradles the guitar neck, and it's not heavy enough to stay put. Even the force from the guitar neck makes it topple over. Unless you glue this thing to the table, or something like that, it's useless, even worse than useless, it's in the way.
Addendum: I raised the rating a bit, after hearing from the distributor/manufacturer ... at least these guys listen.

Fig. 3. Example explanation: Planet Waves Guitar Rest (explanation by Top_Rated_Useful is in grey, that by QUESTER is in green)

Experiments on various product categories show that the QA and the review that are part of a collective explanation are more coherent with each other than those pairings found by the baselines. Review-level and question-level explanations identified by QUESTER are also more consistent with top-rated ones based on helpfulness votes than the baselines.

ACKNOWLEDGMENTS

This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-RP-2021-020).

REFERENCES

[1] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *TOIS*, vol. 22, no. 1, pp. 5–53, 2004.
[2] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
[3] P. Pu, L. Chen, and R. Hu, "A user-centric evaluation framework for recommender systems," in *RecSys*.
[4] Y. Zhang and X. Chen, "Explainable recommendation: A survey and new perspectives," *Foundations and Trends in Information Retrieval*, vol. 14, no. 1, pp. 1–101, 2020.
[5] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Neural attentional rating regression with review-level explanations," in *WWW*, ser. WWW '18, 2018, p. 1583–1592.
[6] H. Liu, Y. Wang, Q. Peng, F. Wu, L. Gan, L. Pan, and P. Jiao, "Hybrid neural recommendation with joint deep representation learning of ratings and reviews," *Neurocomputing*, vol. 374, pp. 77–85, 2020.
[7] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: Understanding rating dimensions with review text," in *RecSys*, ser. RecSys '13, 2013, p. 165–172.
[8] L. Chen, Z. Guan, Q. Xu, Q. Zhang, H. Sun, G. Lu, and D. Cai, "Question-driven purchasing propensity analysis for recommendation," in *AAAI*, vol. 34, no. 01, 2020, pp. 35–42.
[9] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *WSDM*, ser. WSDM '17, 2017, p. 425–434.
[10] Y. Tan, M. Zhang, Y. Liu, and S. Ma, "Rating-boosted latent topics: Understanding users and items with ratings and reviews," in *IJCAI*, ser. IJCAI'16, 2016, p. 2640–2646.
[11] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *WWW*, 2016, pp. 507–517.
[12] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *EMNLP*, 2014, pp. 1532–1543.
[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR (Poster)*, 2015.
[14] C.-Y. Lin and E. Hovy, "Automatic evaluation of summaries using n-gram co-occurrence statistics," in *NAACL*, ser. NAACL '03, 2003, pp. 71–78.
[15] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, pp. 65–72.