Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Summarization-Driven Collaborative Filtering for Explainable Recommendation

REINALD ADRIAN PUGOY[12] (Student Member, IEEE) AND HUNG-YU KAO[1] (Member, IEEE)

[1]Intelligent Knowledge Management Lab, Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan City, Taiwan
[2]Faculty of Information and Communication Studies, University of the Philippines Open University, Los Baños, Philippines

Corresponding Author: Hung-Yu Kao (hykao@mail.ncku.edu.tw)

ABSTRACT The explainability of most review-based recommender systems is limited due to the intrinsic black-box nature of neural networks that has opaqued the models' decision-making processes. This paper presents an explainable, accurate. and novel collaborative filtering framework called SUMMER, which generates extractive or abstractive summary-level explanations for every item and every user. Unlike other explanation types, summary-level explanations closely resemble real-life explanations, and our approach effectively unifies representation and explanation. Text summarization for explainable recommender systems is a largely unexplored area, and we pioneer the integration of collaborative filtering and summarization for explainability. To the best of knowledge, our proposed framework is the first summarization-driven CF that can generate either extractive or abstractive explanations, granting SUMMER a certain degree of flexibility. Also, to the extent of our knowledge, we are the first to emphasize the necessity of reformulating explainability as unsupervised summarization for review-based recommender systems since expecting ground-truth summaries for all items and users in a large dataset is unrealistic. Our experiments demonstrate SUMMER's excellent rating prediction accuracy that is on par with other state-of-the-art approaches. Our explainability study reveals that summary-level explanations are superior and more favorable than other explanation types.

INDEX TERMS Collaborative Filtering, Explainable Recommender Systems, Unsupervised Summarization, Rating Prediction

I. INTRODUCTION

Recommender systems are information filtering systems that play a critical role nowadays. Widely adopted by numerous web applications, they have transformed the manner users discover and evaluate different products and services in light of the ever-increasing volume of online information [1]–[3]. They are becoming ubiquitous as more and more people take advantage of them for leisure activities and transactions such as shopping, watching movies, and reading news articles. For a recommender system to be accurate, it needs to accurately learn user and item representations (i.e., user preferences and item characteristics), which is the primary objective of collaborative filtering (CF). In recommender systems literature, approaches based on CF are considered the most dominant and outstanding models. The earliest CF approaches learned such representations based on user-given numeric ratings, yet employing them is oversimplifying user preferences and item characteristics [4], [5]. The huge quantity of users and items in an online platform results in a highly sparse rating matrix that harms accuracy [6]. To alleviate this, review texts have been utilized to alleviate the said issue. The main advantage of employing reviews as the feature source is that they can cover the inherently multi-dimensional nature of user opinions. Since users can explain their reasons for the ratings that they accord to various items or products, crucial hidden properties may be further uncovered that can consequently influence rating decisions [7]. Hence, reviews contain a large quantity of rich latent information that cannot be otherwise acquired solely from ratings [1].

However, the explainability of most review-based recommender systems is limited. Providing explanations behind predictions is necessary; these could help persuade users to develop further trust in a recommender system and make eventual purchasing decisions [8]–[10]. The intrinsic black-

box nature of neural networks (NN) obscures explainability, and the intricate architecture of hidden layers has opaqued the decision-making processes of neural models [8], [9], [11]. Subsequently, this issue can lead to another dilemma: a trade-off between accuracy and explainability. The most accurate models are usually complicated, non-transparent, and unexplainable [12]. The reverse is also true for explainable and straightforward methods that sacrifice accuracy. Constructing models that are both explainable and accurate is a challenging yet critical research agenda for the machine learning community to ensure that we derive benefits from machine learning fairly and responsibly [8].

Taking these into consideration, recent research efforts have strived to improve this particular aspect of recommender systems. Common types of explanations are word-level and review-level. Review-level explanations are considered to be state-of-the-art; in this setup, the attention mechanism is employed to measure every review's usefulness relative to the item or user embedding [1], [13]. High-scoring reviews are consequently selected to serve as explanations. Furthermore, word-level (or token-level) explanations select words from a local window or textual block [14]–[16]. Not unlike the previous mechanism, top words are chosen due to their high attention weights. Indeed, these two explanation types are side-effects of applying attention to reviews and words, leading to the advantageous formulation of user and item representations. Nevertheless, we argue that both explanation types may not completely resemble real-life explanations. In logic, an explanation is a set of intelligible statements usually constructed to describe and clarify the causes, context, and consequences of objects, events, or phenomena under examination [17]. Based on Table 1, the review-level explanation is identical to the second item review, assuming that it has the higher attention weight. It also inadvertently disregards other possibly useful sentences from other reviews with lower attention scores. In essence, this degenerates into a review selection task. Moreover, even though the word-level explanation contains informative words, it may not be feasible in a real-world recommendation scenario since it typically appears as fragments that may not be intelligible enough [18].

Therefore, we propose and pioneer a novel CF framework called **SUMMER** (or **Summ**arization-Driven Collaborative Filtering for **E**xplainable **R**ecommendation). Our model maintains excellent recommendation performance while uniquely reformulating explainability as a completely unsupervised summarization task. Summarization, a rarely explored area in recommender systems, is a natural language processing task that aims to condense long documents into shorter ones by finding the most relevant information in a given text [19]. Unlike a review-level explanation, a summary-level explanation is expected to retain the most relevant texts across multiple reviews. Other advantages that make it superior and preferred over both review-level and word-level explanations are its coherence, non-redundancy, and readability [20], [21].

TABLE 1. Illustration of different explanation types. A review-level explanation is simply the highest weighted review, and a word-level explanation comprises underlined words with the highest attention scores. Our proposed summary-level explanations closely resemble real-life explanations, wherein the explanation text is derived from the information across multiple reviews.

Reviews Received by an Item (e.g., Printer)
1) This printer has it all. Print, scan, copy, fax and wifi. Wifi makes this printer. No more cables all over the place and no more cluttered desks. Before, if I wanted to print something from my laptop I had to go to the printer and connect the cable. Now I can print over wifi. It prints very beautiful and also scans very high resolutions. Set up was a breeze. Getting other computers to print was also a breeze.
2) First of all, it does it all, and does it well. Print, scan, fax, and photos. Its six-ink system give archival photo prints with long life. This is my first wireless printer, and I have to say, it is a great system: easy to set up, and eliminates that spaghetti-ball of wires. Definitely a big plus. It's fast; very fast. Really cool-looking, and easy to use.
Generated Explanations
• **Word-Level:** First of all, it does it all, and does it well. Print, scan, fax, and photos. Its six-ink system give archival photo prints with long life. This is my first wireless printer, and I have to say, it is a great system: easy to set up, and eliminates that spaghetti-ball of wires. Definitely a big plus. It's fast; very fast. Really cool-looking, and easy to use.
• **Review-Level:** First of all, it does it all, and does it well. Print, scan, fax, and photos. Its six-ink system give archival photo prints with long life. This is my first wireless printer, and I have to say, it is a great system: easy to set up, and eliminates that spaghetti-ball of wires. Definitely a big plus. It's fast; very fast. Really cool-looking, and easy to use.
• **Extractive Summary-Level:** No more cables all over the place and no more cluttered desks. Before, if I wanted to print something from my laptop I had to go to the printer and connect the cable. Its six-ink system give archival photo prints with long life. This is my first wireless printer, and I have to say, it is a great system: easy to set up, and eliminates that spaghetti-ball of wires.
• **Abstractive Summary-Level:** I love this product. It is a great-looking printer and has an answering machine in one place. Setup was easy and I was happy to find this product, but it's a bit less expensive than a good purchase. It is a good value for the money.

In our implementation, SUMMER introduces a summarization layer in a CF architecture. The said layer, which produces summaries for every item and every user, can be powered by either extractive or abstractive summarization. Extractive summarization selects certain salient text segments to comprise the summary, while its abstractive counterpart extensively relies on natural language generation to write the summary [21]. In effect, the summarization layer serves as our encoding mechanism for items and users. The item/user embeddings are essentially pre-trained on the summarization task and later fine-tuned on the rating prediction task. Our novel approach effectively unifies representation and explanation; in other words, a summary both *represents* and *explains* an item or user.

Moreover, we stress that our model performs summarization unsupervised since expecting ground-truth summaries for all items and users in a large dataset is unrealistic. Depending only on labeled datasets limits the domains that a recommender model can be trained on, and obtaining these

targets from scratch is cumbersome, labor-intensive, and time-consuming. This then makes unsupervised summarization appealing because it does not require labeled data [19].

A. CONTRIBUTIONS

These are the major contributions of our study:

1) Summarization for explainable recommender systems is a largely untapped area. We pioneer the integration of summarization and collaborative filtering for explainability. To the best of knowledge, our proposed framework is the first summarization-driven CF that can generate either unsupervised extractive or abstractive summary-level explanations and representations, granting SUMMER a certain degree of flexibility.
2) Also, to the extent of our knowledge, we are the first to emphasize the necessity of unsupervised explainability for review-based recommender systems.
3) Our experiments demonstrate SUMMER's excellent rating prediction accuracy that is on par with or better than other state-of-the-art approaches. Our explainability study reveals that summary-level explanations are more favorable than other explanation types.

II. REVIEW OF RELATED LITERATURE

A. FUNDAMENTAL RECOMMENDER MODELS

Designing a CF model involves two crucial steps: learning user/item representations and modeling user-item interactions based on those representations [22]. One of the fundamental works in the utilization of NN in CF is called neural collaborative filtering (NCF) [23]. NCF, originally implemented for implicit feedback data-driven CF, learns nonlinear, flexible, and more abstractive interactions between users and items by employing multilayer perceptron (MLP) layers as its interaction function. An MLP-based interaction function overcomes the limitations of an inner product-based interaction function. The latter is said to be sub-optimal to learn rich, complicated patterns from real-world data [23].

DeepCoNN is the first deep learning-based model representing users and items from reviews in a joint manner [6]. The model consists of two parallel networks powered by convolutional neural networks (CNN). One network learns user behavior by examining all reviews he has written, and the other network models item properties by exploring all reviews it has received. A shared layer connects these two networks, and factorization machines capture user-item interactions. Another significant model is NARRE, which shares several similarities with DeepCoNN. NARRE is also composed of two parallel CNN-based networks that are uniquely incorporated with the review-level attention mechanism [1]. The said mechanism distinguishes each review's usefulness or contribution based on attention weights. As a side-effect, this also leads to review-level explanations; reviews with the highest attention scores are presented as explanations. These weights are then incorporated into the representations of users and items to enhance embedding quality.

B. RECENT RECOMMENDER MODELS

Other relevant studies include AHN [24], BENEFICT [15], DAML [14], D-Attn [16], DER [25], HUITA [26], MPCN [27], and NCEM [13]. These employ various attention mechanisms to distinguish informative parts of a given data sample, resulting in simultaneous accuracy and explainability improvements. D-Attn integrates global and local attention to score each word to determine its relevance in a review text. DAML utilizes CNN's local and mutual attention to learn review features, and HUITA incorporates a hierarchical, three-tier attention network. MPCN is similar to NARRE, but the former does not rely on convolutional layers. Instead, it introduces a review-by-review pointer-based mechanism that is co-attentive to model user-item relationships. AHN proposes a multi-hierarchical paradigm that recognizes item reviews and user reviews through co-attention. To further enhance user experience, DER produces explanation texts by integrating gated recurrent units (GRU), sentence-level CNN, and personalized attention mechanisms. Furthermore, both BENEFICT and NCEM replace the CNN with a pre-trained BERT model in their parallel user/item networks. The latter is found to be more advantageous since it can fully retain global context and word frequency information, crucial factors that can have consequences on rating prediction accuracy or recommendation performance [28], [29]. For explainability, NCEM similarly adopts NARRE's review-level attention. On the other hand, BENEFICT utilizes BERT's self-attention weights in conjunction with a solution to the maximum subarray problem (MSP). BENEFICT's approach produces an explanation based on a subarray of contiguous tokens with the largest possible sum of self-attention weights. Moreso, this paper is a further generalization of our prior study wherein we proposed a BERT-powered CF model called ESCOFILT, which produces ratio-based extractive summary only for both representation and explanation [18]. The study's results are encouraging as they initially show the promise of summarization for explainability.

In summary, there appears to be a trend; tackling explainability improves prediction and recommendation performance consequentially. While most recommender models address this via attention mechanisms, our proposed model solves this by unifying representation and explanation in the form of summaries.

C. PRINCIPLES OF SUMMARIZATION

The objective of automatic text summarization is to find the most relevant information in a given text document and present it in a more condensed form [21]. A good summary is characterized by its coherence, non-redundancy, and grammatical readability while retaining the most significant contents of the original document [20]. Generally, there are two approaches to producing a summary: extractive and abstractive. In extractive summarization, specific salient sentences are selected from the original document (without any modification) to form the summary [30]. On the other hand, abstractive summarization needs extensive natural lan-

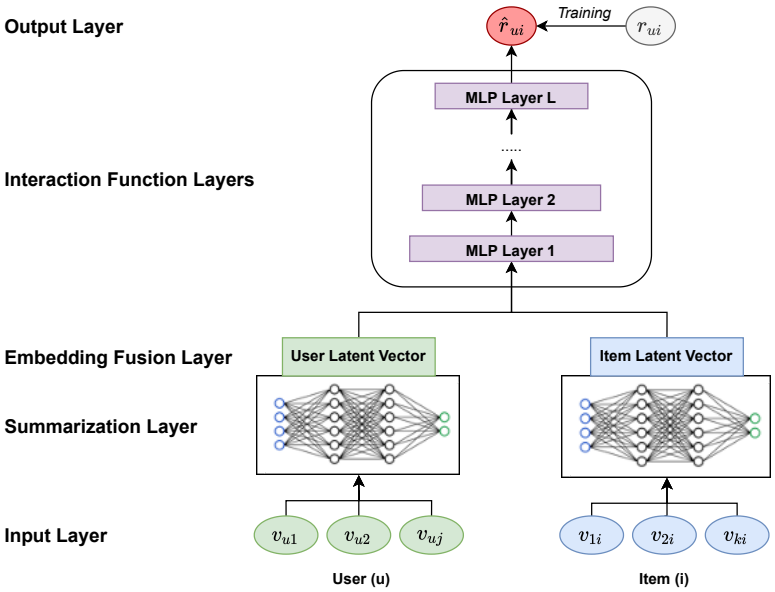

FIGURE 1. The proposed SUMMER framework

guage generation capabilities to rewrite the sentences that comprise the summary. Between the two, extractive methods have gained more attention in the research community [21], and abstractive methods are more difficult as these involve sophisticated techniques typically based on neural networks [19], [31]. Furthermore, summarization approaches can be classified based on the number of source documents: single-document summarization (SDS) and multi-document summarization (MDS). MDS is considered more challenging because of the information overlap among multiple source documents that must be effectively integrated into a concise and comprehensive report.

It is worth noting that the large majority of summarization models are based on supervised learning, thereby requiring extensive collections of labeled training data (i.e., document-summary pairs) [20]. This has practical implications in the real world; labeled datasets are rare and difficult to obtain in the first place, and neural models trained from them do not generalize well to other domains [32]. A simple yet effective unsupervised extractive approach was proposed by Miller [33], which employs BERT to obtain the text embeddings and K-Means clustering to identify sentences closest to the centroids for summary selection. Moreover, Chu and Liu [32] pioneered a state-of-the-art approach called MeanSum, a neural model architecture that performs unsupervised abstractive summarization. The said model consists of an autoencoder where the mean of the input documents' representation decodes to a reasonable summary-document while not leaning on any document-specific features.

Overall, the challenges of summarization also apply to recommender systems' explainability regarding the infeasibility of obtaining large-scale datasets labeled with ground-truth explanations for every item and every user. Thus, unsupervised multi-document summarization is a suitable approach in dealing with CF explainability under our proposed SUMMER framework, wherein a document represents a user review. To grant our model a certain degree of flexibility, it can either be extractive or abstractive, which will be further discussed in the next section of this paper.

III. METHODOLOGY

A. PROBLEM FORMULATION AND OVERVIEW

The training dataset τ consists of N tuples, with the latter indicating the size of the dataset. Each tuple follows this form: (u, i, r_{ui}, v_{ui}) where r_{ui} and v_{ui} respectively denotes the ground-truth rating and review given by user u to item i. Let $RV_u = \{v_{u1}, v_{u2}, ..., v_{uj}\}$ be the set of all j reviews written by user u. Similarly, let $RV_i = \{v_{1i}, v_{2i}, ..., v_{ki}\}$ be the set of all k reviews received by item i. Both RV_u and RV_i are acquired from scanning τ itself row-by-row. SUMMER's input is a user-item pair (u, i) from each tuple in τ. We specifically feed RV_u and RV_i to the model as the initial inputs. The primary output is the predicted rating $\hat{r}_{ui} \in \mathbb{R}$ that user u may give to item i. The rating prediction task can be expressed as:

$$predict(u, i) = (RV_u, RV_i) \rightarrow \hat{r}_{ui} \tag{1}$$

Its corresponding objective function, the mean squared error (MSE), is given below:

$$MSE = \frac{1}{|\tau|} \sum_{u,i \in \tau} (r_{ui} - \hat{r}_{ui})^2 \tag{2}$$

SUMMER's architecture is illustrated in Figure 1. It has two parallel modeling networks that respectively learn

summarization-based user and item representations. For the following subsection of this paper (i.e., "B. Summarization Layer"), we will only discuss the item modeling procedure since it is nearly identical to user modeling, with their inputs as the only difference.

B. SUMMARIZATION LAYER

Through the summarization layer, our model's design is flexible enough to accommodate two possible options for explainability: extractive and abstractive, both of which can effectively represent, explain, and encode users and items. This layer produces the pre-trained summary-level explanation (for every user and every item), which we also call *representative summary*, *representation-explanation*, *explanation-summary*, or simply *summary* in different parts of this paper. This section discusses our unsupervised implementations for either summarization approach.

1) EXTRACTIVE SUMMARIZATION LAYER

The reviews in RV_i are first concatenated together to form a single document. We employ spaCy's Sentencizer, a sentence segmentation tool for splitting the document into individual sentences [34]. The set of all sentences in RV_i is now given by $E_i = \{e_{i1}, e_{i2}, ..., e_{ig}\}$ where g refers to the total number of sentences. Afterward, E_i is fed to a pre-trained BERT model to obtain their corresponding sentence embeddings. This process produces the set of sentence embeddings $E'_i = \{e'_{i1}, e'_{i2}, ..., e'_{ig}\}$, where $E'_i \in \mathbb{R}^{g \times a}$ and a denotes BERT's embedding dimension. The BERT model options can either be standard BERT-Large, wherein the contextualized embeddings can be derived from the penultimate encoder layer [33] or Sentence-BERT, which is based on RoBERTa-Large previously trained on the semantic textual similarity task [35].

Embedding clustering, based on K-Means, is then performed to partition the sentence embeddings in E'_i into K clusters. In our approach, K can be calculated using a hyper-parameter called summary ratio (ϕ), which is the percentage of sentences that shall comprise the actual summary.

$$K_i = \phi_i \times g \tag{3}$$

where K_i is the item K and ϕ_i refers to the item summary ratio. The objective of embedding clustering is to minimize the sum of squared errors (SSE), i.e., the intra-cluster sum of the distances from each sentence to its nearest centroid, given by the following equation [36]:

$$SSE(E'_i) = \sum_{x=1}^{K_i} \sum_{e'_{iy} \in C_x} ||e'_{iy} - c_x||^2 \tag{4}$$

where c_x is the centroid of cluster C_x that is closest to the sentence embedding e'_{iy}. The objective function is optimized for item i by running the assignment and update steps until the cluster centroids stabilize. The assignment step assigns each sentence to a cluster using the shortest distance between the sentence embedding and cluster centroid, provided by the formula below:

$$nc(e'_{iy}) = argmin_{x=1,...,K_i}\{||e'_{iy} - c_x||^2\} \tag{5}$$

where nc is a function that obtains the cluster closest to e'_{iy}. The update step recomputes the cluster centroids based on new assignments from the previous step. This is defined as:

$$c_x = \frac{1}{|C_x|} \sum_{y=1}^{g} \{e'_{iy} | nc(e'_{iy}) = x\} \tag{6}$$

where $|C_x|$ refers to the number of sentences that cluster C_x contains. By introducing clustering, redundant and related sentences are grouped in the same cluster. Sentences closest to each cluster centroid are selected and combined to form the extractive summary. This is expressed as:

$$\begin{aligned} ns(C_x) &= argmin_{y=1,...,g}\{||e'_{iy} - c_x||^2\} \\ XS_i &= [e'_{i,ns(C_1)}, e'_{i,ns(C_2)}, ..., e'_{i,ns(C_{K_i})}] \\ \overline{XS}_i &= \frac{1}{K_i} \sum_{x=1}^{K_i} e'_{i,ns(C_x)} \end{aligned} \tag{7}$$

where ns is a function that returns the nearest sentence to the centroid c_x of cluster C_x, $XS_i \in \mathbb{R}^{K_i \times a}$ is an embedding matrix of the extractive summary sentences, and $\overline{XS}_i \in \mathbb{R}^{1 \times a}$ is the extractive summary embedding of item i.

2) ABSTRACTIVE SUMMARIZATION LAYER

Let $\mathbb{D}$ be the set of all reviews in τ and $|\mathbb{D}|$ be the number of all tuples (i.e., user-item pairs) in τ. We initially have an invertible tokenizer T that maps the reviews in $\mathbb{D}$ to token sequences $T(\mathbb{D})$ from a fixed vocabulary. Also, let $\mathbb{V} \subset T(\mathbb{D})$ denote the tokenized reviews that have a maximum length of H. For item i, given a set of reviews $RV_i \subset \mathbb{V}$, the goal is to produce an explanation-summary $XS_i \in T(\mathbb{D})$ using the same vocabulary.

The abstractive summarization layer contains two key components: the autoencoder and summarization modules. The autoencoder learns representations for each review in the training dataset τ and consequently constrains the generated summaries in its language domain. The encoder ϕ_E maps reviews to real-vector codes denoted by $z_y = \phi_E(v_{yi})$. After processing v one token per every time step, its encoding is expressed by concatenating the LSTM's final hidden and cell states, i.e., $\phi_E(v) = [h, c]$ [37]. Afterward, the decoder LSTM defines a distribution over $\mathbb{V}$ contingent on the latent code $p(v|z_y) = \phi_D(z_y)$. This is accomplished by initializing the decoder's initial state with z_y and training it by teacher-forcing using a standard cross-entropy loss to reconstruct the original reviews. The autoencoder's objective is to minimize the reconstruction loss (REC), which is the collective cross-entropy losses (CE) between the original reviews and their corresponding reconstructed versions:

$$REC(RV_i, \phi_E, \phi_D) = \sum_{y=1}^{k} CE(v_{yi}, \phi_D(\phi_E(v_{yi}))) \tag{8}$$

On the other hand, the summarization module learns to produce explanation-summaries that are semantically similar to the input reviews. The latent codes of the reviews received by item i (i.e., $\{z_1, z_2, z_3, ..., z_k\}$) are integrated by averaging their hidden and cell states in $\bar{z} = [\bar{h}, \bar{c}]$. The joint latent code $\bar{z}$ is decoded by ϕ_D into summary s, which is then later encoded by $\phi_E(s) = [h_s, c_s]$. The encoded summary's hidden state also serves as the item's abstractive summary embedding: $\overline{XS}_i = h_s \in \mathbb{R}^{1\times a}$, where a is the hidden unit size of the encoder.

The process of re-encoding and calculating the similarity loss between the generated summary and its source reviews further constrains the former to be semantically similar to the latter. Regarding this, the following is the objective function that minimizes the similarity loss (SIM) based on the average cosine distance (COS) between the hidden states h_y of each encoded review and $\overline{XS}_i$ of the encoded summary:

$$SIM(RV_i, \phi_E, \phi_D) = \frac{1}{k}\sum_{y=1}^{k} COS(h_y, \overline{XS}_i)) \quad (9)$$

Similar to Chu and Liu's approach [32], the actual summary text is also generated using the Straight Through Gumbel-Softmax strategy [38]. This performs approximated sampling from a categorical distribution, i.e., a softmax over the vocabulary, allowing gradients to be backpropagated through discrete generation.

C. EMBEDDING FUSION LAYER

We also draw certain principles from the traditional latent factor model by incorporating rating-based vectors that depict users and items to a certain extent [1]. These are represented by IV_u and IV_i, both in $\mathbb{R}^{1\times m}$ where m is the dimension of the latent vectors. The hidden vectors are fused with their corresponding summary embeddings. This is facilitated by these fusion levels, illustrated by the following formulas:

$$\begin{aligned} f_u &= (\overline{XS}_u W_u + b_u) + IV_u \\ f_i &= (\overline{XS}_i W_i + b_i) + IV_i \\ f_{ui} &= [f_u, f_i] \end{aligned} \quad (10)$$

where f_u and f_i pertain to the preliminary fusion layers and both are in $\mathbb{R}^{1\times m}$; W_u and W_i are weight matrices in $\mathbb{R}^{a\times m}$; b_u and b_i refer to bias vectors; and $f_{ui} \in \mathbb{R}^{1\times 2m}$ denotes the initial user-item interactions from the third fusion layer.

D. INTERACTION FUNCTION AND RATING PREDICTION

The MLP is essential to model the collaborative filtering effect to learn meaningful non-linear interactions between users and items. An MLP with multiple layers implies a higher degree of non-linearity and flexibility. Similar to the strategy of He et al. [23], SUMMER adopts an MLP with a tower pattern wherein the bottom layer is the widest while every succeeding top layer has fewer neurons. A tower structure enables the MLP to learn more abstractive data features.

TABLE 2. The datasets utilized for our experiments.

Dataset	#Reviews	#Users	#Items
Digital Music	64,706	5,541	3,568
Office Products	53,258	4,905	2,420
Patio, Lawn, & Garden	13,272	1,686	962

Notably, we halve the size of hidden units for each successive higher layer. SUMMER's MLP component is defined as follows:

$$\begin{aligned} h_1 &= ReLU(f_{ui} W_1 + b_1) \\ h_L &= ReLU(h_{L-1} W_L + b_L) \end{aligned} \quad (11)$$

where h_L represents the L-th MLP layer, and W_L and b_L pertain to the L-th layer's weight matrix and bias vector, respectively. We select the rectified linear unit (ReLU) as the activation function since it generally yields better performance than other activation functions [23]. Finally, the MLP's output is projected to one more linear layer to produce the predicted rating:

$$\hat{r}_{ui} = h_L W_{L+1} + b_{L+1} \quad (12)$$

IV. EMPIRICAL EVALUATION

A. RESEARCH QUESTIONS

In this section, we provide the details of our experimental configuration as we aim to answer the following research questions (RQs):

- **RQ1**: Is SUMMER's performance on par with or better than other state-of-the-art baselines?
- **RQ2**: Is summarization an effective encoding mechanism for user and item representations?
- **RQ3**: Are summary-level explanations acceptable to humans in real life?

B. DATASETS, BASELINES, AND EVALUATION METRIC

Table 2 summarizes the three Amazon datasets[1] we used for our experiments. These datasets are 5-core, which implies that every user and every item have a minimum of five reviews [39], [40]. The ratings across all the datasets are in this range: [1, 5]. We further divided a given dataset into training, validation, and test sets using the 80-10-10 split. Then, to compare recommendation performances and validate our model's effectiveness, the following state-of-the-art baselines were executed:

- **DeepCoNN** [6]: The first deep collaborative neural network model that is based on two parallel CNNs to jointly learn user and item features.
- **MPCN** [27]: Akin to NARRE, this CNN-less model employs a new type of dual attention, i.e., co-attention-based pointers, for identifying relevant reviews.
- **NARRE** [1]: Similar to DeepCoNN, it is a neural attentional regression model that integrates two parallel CNNs and the review-level attention mechanism.

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

TABLE 3. Recommendation performance comparison. The best RMSE values are boldfaced.

Model	Digital Music	Office Products	Patio, Lawn, & Garden	Average RMSE
DeepCoNN	0.8904	0.8410	0.9316	0.8876
MPCN	0.9298	0.8487	0.9362	0.9049
NARRE	0.8915	0.8426	0.9539	0.8960
NCF	1.0822	1.0008	1.2359	1.1063
SUMMER-Ext	**0.8831**	**0.8332**	**0.9298**	**0.8820**
SUMMER-Abs	0.8917	0.8356	0.9398	0.8890

TABLE 4. Variants of SUMMER used in our ablation study.

Variant	Item Encoder	User Encoder
SUMMER-1SE	First Sentence Encoding	First Sentence Encoding
SUMMER-IA-U1	Abstractive Summarization	First Sentence Encoding
SUMMER-IX-U1	Extractive Summarization	First Sentence Encoding
SUMMER-I1-UA	First Sentence Encoding	Abstractive Summarization
SUMMER-I1-UX	First Sentence Encoding	Extractive Summarization
SUMMER-5RE	Five Reviews Encoding	Five Reviews Encoding
SUMMER-IA-U5	Abstractive Summarization	Five Reviews Encoding
SUMMER-IX-U5	Extractive Summarization	Five Reviews Encoding
SUMMER-I5-UA	Five Reviews Encoding	Abstractive Summarization
SUMMER-I5-UX	Five Reviews Encoding	Extractive Summarization
SUMMER-Ext	Extractive Summarization	Extractive Summarization
SUMMER-Abs	Abstractive Summarization	Abstractive Summarization

- **NCF** [23]: An interaction-based model that is fundamental in neural recommender systems and uses MLP as the interaction function for the first time.

We calculated each baseline's root mean square error (RMSE) on the test dataset ($\bar{\tau}$) to serve as the evaluation metric. RMSE is a widely accepted metric for assessing a model's rating prediction accuracy [41].

$$RMSE = \sqrt{\frac{1}{|\bar{\tau}|} \sum_{u,i \in \bar{\tau}} (r_{ui} - \hat{r}_{ui})^2} \tag{13}$$

C. EXPERIMENTAL SETTINGS

For the CF component of SUMMER, we operated an exhausting grid search on the number of epochs: [1, 30] and latent vector dimension (m): {128, 200, 220} while fixing the values of the learning rate at 0.006 and number of MLP layers at 4. Moreover, we implemented NCF and also ran a grid search over the number of epochs: [1, 30], latent vector dimension: {128, 200}. For DeepCoNN, MPCN, and NARRE, we employed the extensible NRRec framework[2] and retained the values of the hyperparameters reported in the framework [42]. We performed an exhaustive grid search over the number of epochs: [1, 30] and learning rates: {0.003, 0.004, 0.006}.

All above-mentioned baselines used the same optimizer, Adam, which leverages the power of adaptive learning rates during training [43]. This makes the selection of a learning rate less cumbersome, leading to faster convergence [1]. Without special mention, the models shared the same random seed, batch size (128), and dropout rate (0.5). We selected the model configuration with the lowest RMSE on the validation set.

[2]https://github.com/ShomyLiu/Neu-Review-Rec

It should be noted that we separately trained SUMMER's summarization and rating prediction tasks due to limitations on hardware resources. For the extractive summarization layer, we primarily based its implementation on BERT Extractive Summarizer[3] by Miller [33] using the pre-trained $\text{BERT}_{\text{LARGE}}$ model. Item summary and user summary ratios (i.e., ϕ_i and ϕ_u) were both set to 0.4. Furthermore, for the abstractive summarization layer, we patterned its design after the original MeanSum[4] model of Chu and Liu [32]. The language model, encoders, and decoders were multiplicative LSTMs [44] with hidden unit size of 512, dropout rate of 0.1, word embedding size of 256, and layer normalization [45]. We also used Adam to train the language and summarization models with learning rates of 0.001 and 0.0005, respectively.

V. RESULTS AND DISCUSSION

A. PERFORMANCE COMPARISON

Table 3 summarizes the overall performances of our model and the baselines based on our experiments. These are our findings:

First, SUMMER's extractive version is the top-performing model, acquiring the lowest RMSE scores across all datasets and baselines. This is closely followed by SUMMER's abstractive variant, whose performance is seen comparable to other baselines. These findings prove the effectiveness of summarization as an encoding mechanism for users and items to construct better representations, thereby answering RQ1. Specifically, summarization is helpful in further refining and producing semantically meaningful features that comprise user and item embeddings. Notwithstanding, Ex-

[3]https://github.com/dmmiller612/bert-extractive-summarizer
[4]https://github.com/sosuperic/MeanSum

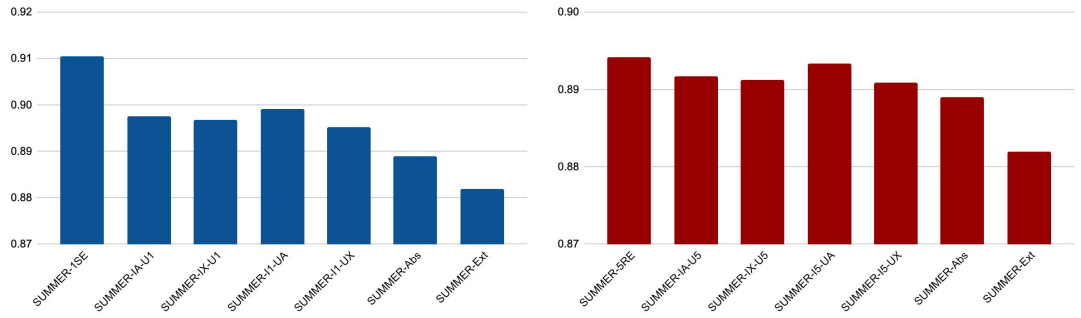

FIGURE 2. Performance comparison of SUMMER variants. The left (a) and right (b) figures illustrate the RMSE scores of SUMMER variants based on 1SE and 5SE replacements, respectively.

tractive SUMMER has a better generalization capability than Abstractive SUMMER.

Second, models that take advantage of review information (i.e., DeepCoNN, MPCN, NARRE, and SUMMER) consistently outperform NCF, the only model based on ratings data alone. This validates the importance of review texts, which are excellent sources of rich information for learning user and item properties. Generally speaking, review-based recommender systems have become reliable in yielding satisfactory and quality rating prediction performance.

B. ABLATION STUDY

This section details our ablation study in order to examine further the efficacy of our proposed summarization layer to encode users and items. In this instance, we separately replaced the user's and item's summarization layer with these non-summarization encoding approaches listed below. Accordingly, we prepared ten variants of SUMMER as shown in Table 4. The rationale behind these approaches is to ensure that they do not resemble SUMMER-generated summaries, aiding us in better conducting and examining the perceived effect of summarization in a CF architecture.

- **First Sentence Encoding (1SE)**: We chose the first sentence of the item (user) review set to represent it. We then projected it to a pre-trained Sentence-BERT model to derive the item (user) embedding.
- **Five Reviews Encoding (5RE)**: This time, we randomly selected five reviews from the item (user) review set. We later fed the concatenated reviews to Sentence-BERT to obtain the embedding of the item (user).

Expectedly, completely removing summarization (as indicated by SUMMER-1SE and SUMMER-5RE) results in the worst accuracy (lowest RMSE scores). It should be noted that the performance immediately improves even if either the item or user component only employs the summarization layer while the remaining component only relies on 1SE or 5RE. Nevertheless, we get the full benefits of summarization if both the item and user components utilize our proposed summarization layer. This is evidenced by our original model's performance (given by SUMMER-Ext and SUMMER-Abs); this answers RQ2.

C. EXPLAINABILITY STUDY

Still in progress...

VI. CONCLUSION AND FUTURE WORK

Soon to finish...

REFERENCES

[1] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Neural attentional rating regression with review-level explanations," in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 1583–1592.

[2] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: understanding rating dimensions with review text," in *Proceedings of the 7th ACM conference on Recommender systems*, 2013, pp. 165–172.

[3] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, p. 5, 2019.

[4] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[5] C. Musto, M. de Gemmis, G. Semeraro, and P. Lops, "A multi-criteria recommender system exploiting aspect-based sentiment analysis of users' reviews," in *Proceedings of the 11th ACM Conference on Recommender Systems*, 2017, pp. 321–325.

[6] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*, 2017, pp. 425–434.

[7] J. Wilson, S. Chaudhury, and B. Lall, "Improving collaborative filtering based recommenders using topic modelling," in *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 01*, 2014, pp. 340–346.

[8] G. Peake and J. Wang, "Explanation mining: Post hoc interpretability of latent factor models for recommendation systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2060–2069.

[9] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?: Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2016, pp. 1135–1144.

[10] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma, "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis," in *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2014, pp. 83–92.

[11] X. Wang, X. He, F. Feng, L. Nie, and T.-S. Chua, "TEM: Tree-enhanced embedding model for explainable recommendation," in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 1543–1552.

[12] Y. Zhang and X. Chen, "Explainable recommendation: A survey and new perspectives," *arXiv preprint arXiv:1804.11192*, 2018.

[13] X. Feng and Y. Zeng, "Neural collaborative embedding from reviews for recommendation," *IEEE Access*, vol. 7, pp. 103 263–103 274, 2019.

[14] D. Liu, J. Li, B. Du, J. Chang, and R. Gao, "DAML: Dual attention mutual learning between ratings and reviews for item recommendation,"

in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 344–352.
[15] R. A. Pugoy and H.-Y. Kao, "BERT-based neural collaborative filtering and fixed-length contiguous tokens explanation," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, 2020, pp. 143–153.
[16] S. Seo, J. Huang, H. Yang, and Y. Liu, "Interpretable convolutional neural networks with dual local and global attention for review rating prediction," in *Proceedings of the 11th ACM Conference on Recommender Systems*, 2017, pp. 297–305.
[17] J. Drake, *Introduction to Logic*. Scientific e-Resources, 2018.
[18] R. A. Pugoy and H.-Y. Kao, "Unsupervised extractive summarization-based representations for accurate and explainable collaborative filtering," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Aug. 2021, pp. 2981–2990.
[19] J. Zhao, M. Liu, L. Gao, Y. Jin, L. Du, H. Zhao, H. Zhang, and G. Haffari, "Summpip: Unsupervised multi-document summarization with sentence graph compression," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 1949–1952.
[20] R. R. Chowdhury, M. T. Nayeem, T. T. Mim, M. S. R. Chowdhury, and T. Jannat, "Unsupervised abstractive summarization of bengali text documents," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021, pp. 2612–2619.
[21] M. T. Nayeem, T. A. Fuad, and Y. Chali, "Abstractive unsupervised multi-document summarization using paraphrastic sentence fusion," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 1191–1204.
[22] X. He, X. Du, X. Wang, F. Tian, J. Tang, and T.-S. Chua, "Outer product-based neural collaborative filtering," *arXiv preprint arXiv:1808.03912*, 2018.
[23] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 173–182.
[24] X. Dong, J. Ni, W. Cheng, Z. Chen, B. Zong, D. Song, Y. Liu, H. Chen, and G. de Melo, "Asymmetrical hierarchical networks with attentive interactions for interpretable review-based recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 7667–7674.
[25] X. Chen, Y. Zhang, and Z. Qin, "Dynamic explainable recommendation based on neural attentive models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 53–60.
[26] C. Wu, F. Wu, J. Liu, and Y. Huang, "Hierarchical user and item representation with three-tier attention for recommendation," 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)*, 2019, pp. 1818–1826.
[27] Y. Tay, A. T. Luu, and S. C. Hui, "Multi-pointer co-attention networks for recommendation," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2309–2318.
[28] M. T. Pilehvar and J. Camacho-Collados, "WiC: the word-in-context dataset for evaluating context-sensitive meaning representations," 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)*, 2019, pp. 1267–1273.
[29] Q. Wang, S. Li, and G. Chen, "Word-driven and context-aware review modeling for recommendation," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 1859–1862.
[30] R. Nallapati, B. Zhou, and M. Ma, "Classify or select: Neural architectures for extractive document summarization," *arXiv preprint arXiv:1611.04244*, 2016.
[31] H. Zhang, J. Xu, and J. Wang, "Pretraining-based natural language generation for text summarization," *arXiv preprint arXiv:1902.09243*, 2019.
[32] E. Chu and P. Liu, "Meansum: a neural model for unsupervised multi-document abstractive summarization," in *International Conference on Machine Learning*. PMLR, 2019, pp. 1223–1232.
[33] D. Miller, "Leveraging bert for extractive text summarization on lectures," *arXiv preprint arXiv:1906.04165*, 2019.
[34] S. Gupta and K. Nishu, "Mapping local news coverage: Precise location extraction in textual news content using fine-tuned bert based language model," in *Proceedings of the 4th Workshop on Natural Language Processing and Computational Social Science*, 2020, pp. 155–162.
[35] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.
[36] S. Xia, D. Peng, D. Meng, C. Zhang, G. Wang, E. Giem, W. Wei, and Z. Chen, "A fast adaptive k-means with no bounds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
[37] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[38] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," *arXiv preprint arXiv:1611.00712*, 2016.
[39] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel, "Image-based recommendations on styles and substitutes," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015, pp. 43–52.
[40] R. He and J. McAuley, "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*, 2016, pp. 507–517.
[41] H. Steck, "Evaluation of recommendations: rating-prediction and ranking," in *Proceedings of the 7th ACM Conference on Recommender systems*, 2013, pp. 213–220.
[42] H. Liu, F. Wu, W. Wang, X. Wang, P. Jiao, C. Wu, and X. Xie, "NRPA: Neural recommendation with personalized attention," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 1233–1236.
[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[44] B. Krause, L. Lu, I. Murray, and S. Renals, "Multiplicative lstm for sequence modelling," *arXiv preprint arXiv:1609.07959*, 2016.
[45] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

REINALD ADRIAN PUGOY (Student Member, IEEE) received his BS and MS degrees in Computer Science from the University of the Philippines Los Baños in 2008 and 2011, respectively. He is currently a PhD candidate at the Department of Computer Science and Information Engineering (CSIE), National Cheng Kung University (NCKU), Taiwan. He is also a faculty member (currently on study leave) at the Faculty of Information and Communication Studies, University of the Philippines Open University. His research interests include recommender systems, natural language processing (NLP), machine learning, and data mining.

HUNG-YU KAO received his BS and MS degrees in Computer Science from the National Tsing Hua University, Taiwan in 1994 and 1996, respectively. In 2003, he received his PhD degree from the Electrical Engineering Department, National Taiwan University. He is currently a professor at the CSIE department of NCKU. Dr. Kao is also the chair of IEEE CIS Tainan Chapter. He was a postdoctoral fellow at the Institute of Information Science, Academia Sinica, Taiwan from 2003 to 2004. His research interests include NLP, web information retrieval and extraction, knowledge management, data mining, social network analysis, and bioinformatics.