Summarization-Driven Collaborative Filtering for Explainable Recommendation

Reinald Adrian Pugoy
University of the Philippines Open University
Los Baños, Laguna, Philippines
rdpugoy@up.edu.ph

Hung-Yu Kao
National Cheng Kung University
Tainan City, Taiwan
hykao@mail.ncku.edu.tw

ABSTRACT

Neural review-based recommender systems often lack explainability due to the black-box nature of neural networks. This paper introduces SUMMER, a novel, accurate, and explainable collaborative filtering (CF) framework. SUMMER generates summary-level explanations for each item and user, mirroring the style of real-life explanation texts. This integration of summarization into the CF architecture not only improves explainability but also enhances the encoding of users and items, boosting recommendation performance. SUMMER is the first summarization-driven CF model capable of generating both extractive and abstractive explanations, offering flexibility in explanation generation. We further argue for reformulating explainability as unsupervised summarization, recognizing the impracticality of obtaining ground-truth explanations for every item and user. Our experiments demonstrate SUMMER's strong rating prediction accuracy, comparable to other state-of-the-art approaches. Moreover, our explainability study reveals a user preference for extractive summary-level explanations.

KEYWORDS

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

ACM Reference Format:
Reinald Adrian Pugoy and Hung-Yu Kao. 2024. Summarization-Driven Collaborative Filtering for Explainable Recommendation. In *Proceedings of Philippine Computing Science Congress (PCSC2024).* Laguna, Philippines, 8 pages.

1 INTRODUCTION

Recommender systems have become indispensable tools in navigating the vast landscape of online information. Widely integrated into web applications, they revolutionize how users discover and assess products and services across various domains, from shopping to entertainment and news consumption [3, 35]. Collaborative filtering (CF) lies at the heart of these systems, aiming to accurately capture user preferences and item characteristics. While early CF models relied solely on numeric ratings, this approach oversimplifies the nuanced nature of user preferences and suffers from sparse rating matrices, impairing accuracy [16, 21, 38]. To address these challenges, researchers have turned to review texts as a valuable source of information. By leveraging user-given reviews that discuss the rationale behind the ratings, recommender systems can uncover latent properties and dimensions of user opinions that are not captured by ratings alone [31]. Reviews offer a wealth of rich, multidimensional insights that cannot be otherwise acquired solely from ratings [3].

However, most neural review-based recommender systems lack explainability, which is crucial for user trust and decision-making [23, 25, 37]. The inherent opacity of neural networks, often referred to as *black-boxes*, creates a a dilemma: a trade-off between

Table 1: Illustration of the different explanation types. A review-level explanation is simply the highest weighted review, and a word-level explanation is composed of underlined words with the highest attention scores. Our proposed summary-level explanations closely resemble real-life explanations by deriving information from 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.

PCSC2024, May 2024, Laguna, Philippines

accuracy and explainability. [23, 25, 30]. The most accurate models often suffer from complexity and a lack of explainability [36]. Conversely, simple, explainable methods may compromise accuracy. Striking a balance between explainability and accuracy poses a significant challenge. Constructing models that are both explainable and accurate is a critical research agenda for the machine learning community to ensure that we derive benefits from machine learning fairly and responsibly [23].

To improve user trust and understanding, recent research has explored various explainability methods for recommender systems. Common approaches include review-level and word-level explanations. Review-level explanations, utilizing attention mechanisms to select high-scoring reviews, are considered state-of-the-art [3, 8]. Word-level explanations, which select top words based on attention weights, offer another strategy [26]. Nevertheless, both types may not fully resemble real-life explanations; as an illustration, in 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 reviews with lower attention scores. In essence, this degenerates into a review selection task. Moreover, while a word-level explanation highlights relevant terms, its fragmented nature may hinder real-world recommendation scenarios, as it may lack overall coherence and intelligibility for users.

Therefore, we propose and pioneer a novel CF framework called **SUMMER** (derived from **Summ**arization-Driven Collaborative Filtering for **E**xplainable **R**ecommendation). Our model offers competitive recommendation performance and contributes to explainability research by exploring a less-investigated approach: treating it as an unsupervised summarization task within recommender systems. 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 preferred are its coherence, non-redundancy, and readability [4, 22]. In our implementation, SUMMER integrates a summarization layer into a CF architecture. This layer generates either extractive (selecting salient text segments) or abstractive (rephrasing with natural language generation) summaries for each item and user. The summarization layer acts as an encoding mechanism, with item/user embeddings pre-trained on the summarization task and fine-tuned for rating prediction. This novel approach unifies representation and explanation – the summary both *represents* and *explains* an item (or user). Importantly, our model performs unsupervised summarization, as expecting ground-truth summaries for large datasets is unrealistic and obtaining them manually is cumbersome. This lack of reliance on labeled data makes the approach especially appealing.

1.1 Contributions

These are the major contributions of our study:

(1) We pioneer the integration of summarization and collaborative filtering for explainability. To the best of our knowledge, SUMMER is the first CF model capable of generating either extractive or abstractive explanations, offering a certain degree of flexibility.
(2) To our knowledge, we are the first to emphasize that reformulating explainability as unsupervised summarization is necessary to address the impracticality of ground-truth explanations.
(3) Our experiments demonstrate SUMMER's competitive rating prediction accuracy, aligning with or surpassing other state-of-the-art methods. In the context of explainability, our study suggests a potential preference for summary-level explanations, with extractive summaries being particularly well-received.
(4) This study additionally explores the impact of explanation type (extractive or abstractive) on both recommendation performance and real-life acceptability.

2 REVIEW OF RELATED LITERATURE

Designing a collaborative filtering (CF) model involves two key steps: learning user/item representations and modeling user-item interactions based on those representations [11]. A fundamental work in this domain is neural collaborative filtering (NCF), which utilizes multilayer perceptron (MLP) layers to learn flexible interactions between users and items [12]. NCF overcomes the limitations of inner product-based interaction functions, enabling it to capture rich patterns in real-world data. DeepCoNN is the first model to jointly represent users and items using reviews, employing convolutional neural networks (CNN) in parallel networks connected by a shared layer [38]. NARRE, similar to DeepCoNN, incorporates review-level attention mechanisms to enhance embedding quality and provide review-level explanations [3].

Other notable studies include AHN, HUITA, MPCN, and NCEM, which employ various attention mechanisms to improve accuracy and explainability [7, 8, 28, 32]. These models integrate attention mechanisms differently to discern informative parts of data samples, leading to enhanced performance. 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 user and item reviews through co-attention. NCEM replaces the CNN with a pre-trained BERT model in its parallel user/item networks. Incorporating BERT 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 [29]. 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.

On the principles of text summarization, two main approaches exist: extractive, which selects important sentences as they are, and abstractive, which rewrites sentences. Extractive methods are more commonly researched [22], while abstractive methods require advanced natural language generation techniques [34]. Summarization can also be categorized by the number of source documents: single-document summarization (SDS) and multi-document summarization (MDS), with MDS being more challenging due to integrating information from multiple sources [4]. Most summarization

models rely on supervised learning, necessitating labeled training data, which is often scarce and leads to poor generalization across domains [4, 5]. Miller proposed an unsupervised extractive method using BERT embeddings and K-Means clustering for sentence selection [20]. Chu and Liu introduced MeanSum, an unsupervised abstractive summarization model based on an autoencoder architecture [5].

The challenges inherent in summarization also apply to explainable recommender systems, where obtaining labeled datasets with ground-truth explanations is impractical. This makes unsupervised multi-document summarization particularly valuable. Within the proposed SUMMER framework, each document represents a user review, providing flexibility in adopting either extractive or abstractive methods.

3 METHODOLOGY

3.1 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-derived user and item representations. For the following subsection of this paper (i.e., *3.2 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.

3.2 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*, or *explanation-summary* in different parts of this paper. This section discusses our unsupervised implementations for either summarization approach.

3.2.1 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 [9]. 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 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 choices can either be standard BERT-Large, wherein the contextualized embeddings can be derived from the penultimate encoder layer [20] or Sentence-BERT, which is based on RoBERTa-Large previously trained on the semantic textual similarity task [24]. 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 hyperparameter called summary ratio (ϕ), which is the percentage of sentences that shall comprise the actual summary.

$$K = \phi \times g \tag{3}$$

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 [33]:

$$SSE_i = \sum_{x=1}^{K} \sum_{e'_{iy} \in C_x} ||e'_{iy} - c_x||^2 \tag{4}$$

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:

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

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:

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

where ns is a function that returns the nearest sentence to the centroid c_x of cluster C_x, $XS_i \in \mathbb{R}^{K \times a}$ 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.

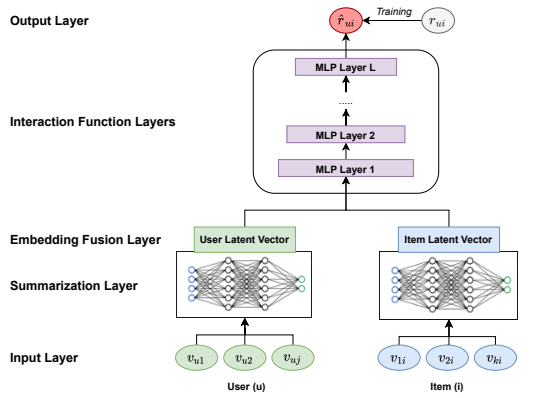

Figure 1: The proposed SUMMER framework.

3.2.2 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., $\phi_E(v) = [h, c]$ [13]. 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:

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

Similar to Chu and Liu's approach [5], the actual summary text is also generated using the Straight Through Gumbel-Softmax strategy. This performs approximated sampling from a categorical distribution, i.e., a softmax over the vocabulary, allowing gradients to be backpropagated through discrete generation.

3.3 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 [3]. 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:

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

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.

3.4 Interaction Function and Rating Prediction

The MLP is essential to model the CF effect to learn meaningful 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. [12], 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. Notably, we halve the size of hidden units for each successive higher layer. SUMMER's MLP component is defined as follows:

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

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 choose the rectified linear unit (ReLU) as the activation function since it generally yields better performance than other activation functions [12]. Finally, the MLP's output is projected to one more linear layer to produce the predicted rating:

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

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

Table 3: Variants of SUMMER used in our ablation study.

Variant	Type	Item Encoder	User Encoder
SUMMER-1SE	Null	First Sentence	First Sentence
SUMMER-IA-U1	Hybrid	Abstractive Summ.	First Sentence
SUMMER-IX-U1	Hybrid	Extractive Summ.	First Sentence
SUMMER-I1-UA	Hybrid	First Sentence	Abstractive Summ.
SUMMER-I1-UX	Hybrid	First Sentence	Extractive Summ.
SUMMER-5RE	Null	Five Reviews	Five Reviews
SUMMER-IA-U5	Hybrid	Abstractive Summ.	Five Reviews
SUMMER-IX-U5	Hybrid	Extractive Summ.	Five Reviews
SUMMER-I5-UA	Hybrid	Five Reviews	Abstractive Summ.
SUMMER-I5-UX	Hybrid	Five Reviews	Extractive Summ.
SUMMER-Ext	Original	Extractive Summ.	Extractive Summ.
SUMMER-Abs	Original	Abstractive Summ.	Abstractive Summ.

4 EMPIRICAL EVALUATION

4.1 Research Questions

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

- **RQ1**: How does SUMMER's rating prediction accuracy compare to other state-of-the-art baselines?
- **RQ2**: To what extent does summarization (as an encoding mechanism) improve the effectiveness of user and item representations?
- **RQ3**: How are summary-based explanations perceived by humans in real life?
- **RQ4**: How do extractive and abstractive representation-explanations compare in the following:
  - **RQ4a**: Their impact on recommendation performance?
  - **RQ4b**: Their real-life acceptability to users?

4.2 Datasets, Baselines, and Evaluation Metric

Table 2 summarizes the three Amazon datasets[1] we utilized in our experiments. These datasets are 5-core, implying that every user and every item have a minimum of five reviews [10, 19]. The ratings across all the datasets are in the range of one to five. 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 used:

- **DeepCoNN** [38]: The first deep collaborative neural network model that is based on two parallel CNNs to jointly learn user and item features.
- **MPCN** [28]: Akin to NARRE, MPCN implements a new type of dual attention for identifying relevant reviews.
- **NARRE** [3]: Similar to DeepCoNN, it is a neural attentional regression model that integrates two parallel CNNs and the review-level attention mechanism.
- **NCF** [12]: An interaction-based model that is fundamental in neural recommender systems; the first to introduce the MLP as the interaction function.

For the evaluation metric, we calculated each baseline's root mean square error (RMSE) on the test dataset ($\bar{\tau}$). RMSE is a widely accepted metric for assessing a model's accuracy and recommendation performance [27].

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

4.3 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] and latent vector dimension: {128, 200}. For DeepCoNN, MPCN, and NARRE, we availed the extensible NRRec framework[2] and retained the hyperparameters' values reported in the framework [18]. We performed an exhaustive grid search over the number of epochs: [1, 30] and learning rates: {0.003, 0.004, 0.006}.

All the above-mentioned baselines used the same optimizer, Adam, which leverages the power of adaptive learning rates during training [15]. This makes the selection of learning rates less cumbersome, leading to faster convergence [3]. 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. We then 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 [20] using the pre-trained $\text{BERT}_{\text{LARGE}}$ model. The summary ratio (ϕ) was set to 0.4. For the abstractive summarization layer, we patterned its design after the original MeanSum[4] model of Chu and Liu [5]. The language model, encoders, and decoders were multiplicative LSTMs [17] with hidden unit size of 512, dropout rate of 0.1, word embedding size of 256, and layer normalization [1]. We also took advantage of the Adam optimizer to train the language and summarization models with learning rates of 0.001 and 0.0005, respectively.

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

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

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

[4] https://github.com/sosuperic/MeanSum

Table 4: 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

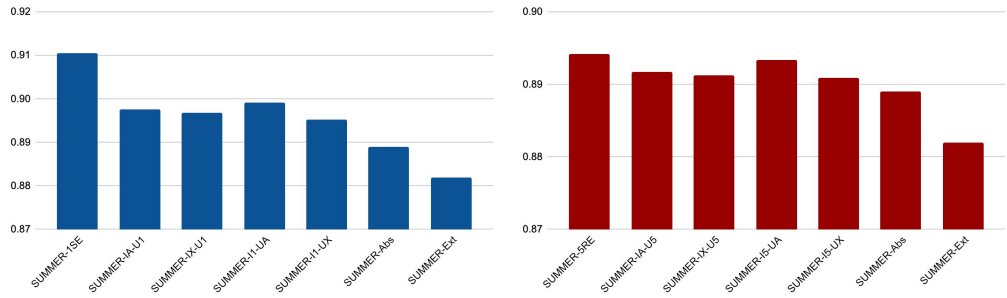

Figure 2: Performance comparison of SUMMER variants. The left (a) and right (b) figures illustrate the RMSE scores of SUMMER variants based on 1SE and 5RE ablations, respectively.

5 PREDICTION RESULTS AND DISCUSSION

5.1 Performance Comparison

After conducting our experiments, we found out that 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 comparable to other baselines. These observations answer RQ1; our findings prove that integrating summarization in a CF architecture is effective, subsequently resulting in better representations and accurate recommendations. Specifically, summarization is helpful in further refining and producing semantically meaningful features with finer granularity and fewer redundancies to comprise user and item embeddings. Notwithstanding, extractive SUMMER appears to have a better generalization capability than abstractive SUMMER; this addresses RQ4a.

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

5.2 Ablation Study

5.2.1 Configuration. In order to examine further the efficacy of our proposed summarization layer for encoding users and items, we separately replaced the user's and item's summarization layer with non-summarization encoding approaches (listed below). The rationale behind these approaches is that to examine better the perceived overall effect of summarization in CF, we must ensure that the ablated component is replaced by an encoding that does not resemble SUMMER-generated summaries.

- **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)**: 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).

Accordingly, as described in Table 3, we prepared ten other variants of SUMMER that utilize different combinations of encoding mechanisms for the user and item components. These are different from the *original variants* (i.e., SUMMER-Ext and SUMMER-Abs); they are non-original and can be categorized into the *null variant* or *hybrid variant*. A null variant, such as SUMMER-1SE and SUMMER-5RE, completely replaces the summarization layer with its corresponding encoding mechanism (i.e., 1SE or 5RE). On the other hand, for a hybrid variant, either the user or item component maintains the summarization layer while the other ablated component employs 1SE or 5RE.

5.2.2 Analysis. The results of our ablation experiments are depicted in Figure 2. Completely removing summarization expectedly results in the least accurate performances, as evidenced by SUMMER-1SE and SUMMER-5RE receiving the lowest RMSE scores. The performance immediately improves even if only either component (item or user) takes advantage of summarization. Hence, it is noticeable that the RMSE values of hybrid variants are significantly

Table 5: Acceptability comparison between extractive and abstractive summary-level explanations. The best mean values are boldfaced. The symbol ** indicates that the difference is statistically significant (at p < 0.05).

Quality	Extractive	Abstractive
Coherence	**3.33**	3.27
Focus	**3.31**	3.27
Grammaticality	**3.25**	3.20
Non-Redundancy	3.15	**3.18**
Referential Clarity	**3.59**	3.16
Usefulness	**3.50****	3.20

better than the null variants. Lastly, the full benefits of summarization are realized when both item and user components utilize our proposed summarization layer. Across all variants, SUMMER-Ext and SUMMER-Abs have the two best RMSE scores; this conclusively answers RQ2.

6 EXPLAINABILITY STUDY

6.1 Listwise Evaluation of Helpfulness

It is intuitive to perform a listwise evaluation (i.e., a ranking-based analysis) to examine whether explanations generated by SUMMER are on par with review-level explanations in real life, considering that the latter have been popular in recommender systems literature. In this regard, we generated a tuple of explanation texts (SUMMER's extractive and abstractive summaries and NARRE's review-level texts) for 30 items, totaling 90 explanations to be assessed. For each item explanation tuple, we instructed four English-speaking human judges to rank them according to helpfulness. A statement is considered helpful if it can aid the viewing user both to know a product better and make a future purchasing (or non-purchasing) decision. Additionally, we determined the strength of inter-judge ranking agreement by utilizing Fleiss' Kappa (κ) wherein 0 refers to a random agreement, and 1 denotes a perfect agreement [2, 14].

Figure 3 shows a clear tendency toward extractive summaries, ranked first for nearly 60% of items. Review-level explanations follow (31%), while abstractive summaries are favored least (ranked first 10% and ranked last 63%). With a fair inter-judge agreement (Kappa value of 0.25), this highlights the superiority of extractive summaries in real-life acceptability (RQ3), exceeding review-level explanations. The lower ranking of abstractive summaries suggests potential inaccuracies or irrelevancies introduced by the generative model (partly answering RQ4b).

6.2 Acceptability Comparison

To fully address RQ4b, it is reasonable to conduct a detailed comparison between extractive and abstractive explanations based on their inherent qualities. In the absence of a ground-truth summary, we propose assessing the real-life acceptability of explanations based on the following summarization criteria, measured on a scale of 1 (poor) to 5 (excellent). We adopted the first five qualities from Dang [6], while we added the sixth dimension of usefulness.

(1) **Coherence**: Should be well-structured and well-organized. It should build from sentence to sentence to a coherent body of information about an item/product/topic.
(2) **Focus**: Should only contain information that is related to the rest of the summary.
(3) **Grammaticality**: No obvious ungrammatical sentences (e.g., fragments, missing components) that make the text difficult to read.
(4) **Non-Redundancy**: No unnecessary repetition in the summary.
(5) **Referential Clarity**: Easy to identify who or what the noun phrases and pronouns are referring to in the summary.
(6) **Usefulness**: Provides useful information to help users decide in making a purchasing decision.

Considering this, we produced a total of 60 item explanations, 30 each from SUMMER's extractive and abstractive types. We likewise asked four English-speaking judges to independently evaluate the explanations according to the earlier-mentioned summarization criteria, and we ascertained their agreement level by utilizing Fleiss' Kappa. Furthermore, we employed the t-test to determine whether the difference between the two explanation types is statistically significant for each summarization quality.

Table 5 reveals that extractive explanations outscore abstractive explanations in five out of six quality metrics. Specifically, they are most grammatical, coherent, and useful; they also provide the best focus and clarity. Their strongest aspects are referential clarity and usefulness. The former has a mean score of 3.59 and a κ value of nearly 0.1, indicating a slight agreement between a pair of judges. The latter has a mean score of 3.50 and a p-value of about 0.03, implying that the difference is statistically significant from abstractive explanations' usefulness. Interestingly, abstractive summaries are less redundant than their extractive counterparts (with a mean score of 3.18).

Overall, these findings demonstrate a strong preference for extractive explanations over abstractive ones in real-life settings, addressing RQ4b. Extractive explanations provide clear, well-structured guidance to help users understand items and make informed purchasing decisions.

7 CONCLUSION AND FUTURE WORK

We have successfully implemented SUMMER, a novel summarization-driven CF model that is both accurate and explainable. By integrating summarization as an encoding mechanism within a CF architecture, we generate semantically rich user and item representations, resulting in competitive recommendation performance. Our experiments confirm that SUMMER's accuracy aligns with or surpasses other state-of-the-art approaches. Moreover, our explainability study demonstrates the real-life favorability of extractive summary-level explanations.

Nevertheless, our study highlights several avenues for further research. Expanding human evaluation with a larger pool of judges and thoroughly analyzing factors contributing to varying inter-rater agreement offers one valuable direction. Additionally, a deeper investigation of the accuracy-explainability trade-off is crucial. Analyzing specific cases where less helpful explanations might also impact prediction accuracy could reveal areas for improving both

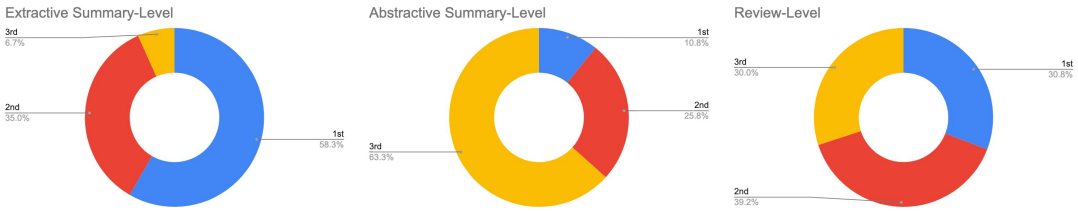

Figure 3: Distribution of the judges' given helpfulness rankings for listwise evaluation.

the informativeness of explanations and the overall model performance.

REFERENCES

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[2] Ria Mae Borromeo and Motomichi Toyama. 2015. Automatic vs. crowdsourced sentiment analysis. In *Proceedings of the 19th International Database Engineering & Applications Symposium*. 90–95.

[3] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural attentional rating regression with review-level explanations. In *Proceedings of the 2018 World Wide Web Conference*. 1583–1592.

[4] Radia Rayan Chowdhury, Mir Tafseer Nayeem, Tahsin Tasnim Mim, Md Saifur Rahman Chowdhury, and Taufiqul Jannat. 2021. 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*. 2612–2619.

[5] Eric Chu and Peter Liu. 2019. Meansum: a neural model for unsupervised multi-document abstractive summarization. In *International Conference on Machine Learning*. PMLR, 1223–1232.

[6] Hoa Trang Dang. 2006. DUC 2005: Evaluation of question-focused summarization systems. In *Proceedings of the Workshop on Task-Focused Summarization and Question Answering*. 48–55.

[7] Xin Dong, Jingchao Ni, Wei Cheng, Zhengzhang Chen, Bo Zong, Dongjin Song, Yanchi Liu, Haifeng Chen, and Gerard de Melo. 2020. Asymmetrical hierarchical networks with attentive interactions for interpretable review-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 7667–7674.

[8] Xingjie Feng and Yunze Zeng. 2019. Neural Collaborative Embedding From Reviews for Recommendation. *IEEE Access* 7 (2019), 103263–103274.

[9] Sarang Gupta and Kumari Nishu. 2020. 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*. 155–162.

[10] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*. 507–517.

[11] Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jinhui Tang, and Tat-Seng Chua. 2018. Outer product-based neural collaborative filtering. *arXiv preprint arXiv:1808.03912* (2018).

[12] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. 173–182.

[13] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[14] Selim Kiliç. 2015. Kappa test. *Psychiatry and Behavioral Sciences* 5, 3 (2015), 142.

[15] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[16] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[17] Ben Krause, Liang Lu, Iain Murray, and Steve Renals. 2016. Multiplicative LSTM for sequence modelling. *arXiv preprint arXiv:1609.07959* (2016).

[18] Hongtao Liu, Fangzhao Wu, Wenjun Wang, Xianchen Wang, Pengfei Jiao, Chuhan Wu, and Xing Xie. 2019. NRPA: Neural Recommendation with Personalized Attention. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1233–1236.

[19] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 43–52.

[20] Derek Miller. 2019. Leveraging BERT for extractive text summarization on lectures. *arXiv preprint arXiv:1906.04165* (2019).

[21] Cataldo Musto, Marco de Gemmis, Giovanni Semeraro, and Pasquale Lops. 2017. A Multi-criteria Recommender System Exploiting Aspect-based Sentiment Analysis of Users' Reviews. In *Proceedings of the 11th ACM Conference on Recommender Systems*. 321–325.

[22] Mir Tafseer Nayeem, Tanvir Ahmed Fuad, and Yllias Chali. 2018. Abstractive unsupervised multi-document summarization using paraphrastic sentence fusion. In *Proceedings of the 27th International Conference on Computational Linguistics*. 1191–1204.

[23] Georgina Peake and Jun Wang. 2018. 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*. 2060–2069.

[24] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[25] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. 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*. 1135–1144.

[26] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Proceedings of the 11th ACM Conference on Recommender Systems*. 297–305.

[27] Harald Steck. 2013. Evaluation of recommendations: rating-prediction and ranking. In *Proceedings of the 7th ACM Conference on Recommender systems*. 213–220.

[28] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Multi-pointer co-attention networks for recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2309–2318.

[29] Qianqian Wang, Si Li, and Guang Chen. 2018. Word-driven and context-aware review modeling for recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 1859–1862.

[30] Xiang Wang, Xiangnan He, Fuli Feng, Liqiang Nie, and Tat-Seng Chua. 2018. TEM: Tree-enhanced embedding model for explainable recommendation. In *Proceedings of the 2018 World Wide Web Conference*. 1543–1552.

[31] Jobin Wilson, Santanu Chaudhury, and Brejesh Lall. 2014. 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*. 340–346.

[32] Chuhan Wu, Fangzhao Wu, Junxin Liu, and Yongfeng Huang. 2019. 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)*. 1818–1826.

[33] Shuyin Xia, Daowan Peng, Deyu Meng, Changqing Zhang, Guoyin Wang, Elisabeth Giem, Wei Wei, and Zizhong Chen. 2020. A fast adaptive K-means with no bounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[34] Haoyu Zhang, Jianjun Xu, and Ji Wang. 2019. Pretraining-based natural language generation for text summarization. *arXiv preprint arXiv:1902.09243* (2019).

[35] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)* 52, 1 (2019), 5.

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

[37] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 83–92.

[38] Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. 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*. 425–434.