information

Article

A Joint Summarization and Pre-Trained Model for Review-Based Recommendation

Yi Bai, Yang Li * and Letian Wang

College of Information and Computer Engineering, Northeast Forestry University, Harbin 150004, China; by@nefu.edu.cn (Y.B.); letian@nefu.edu.cn (L.W.)
* Correspondence: yli@nefu.edu.cn

Abstract: Currently, reviews on the Internet contain abundant information about users and products, and this information is of great value to recommendation systems. As a result, review-based recommendations have begun to show their effectiveness and research value. Due to the accumulation of a large number of reviews, it has become very important to extract useful information from reviews. Automatic summarization can capture important information from a set of documents and present it in the form of a brief summary. Therefore, integrating automatic summarization into recommendation systems is a potential approach for solving this problem. Based on this idea, we propose a joint summarization and pre-trained recommendation model for review-based rate prediction. Through automatic summarization and a pre-trained language model, the overall recommendation model learns a fine-grained summary representation of the key content as well as the relationships between words and sentences in each review. The review summary representations of users and items are finally incorporated into a neural collaborative filtering (CF) framework with interactive attention mechanisms to predict the rating scores. We perform experiments on the Amazon dataset and compare our method with several competitive baselines. Experimental results show that the performance of the proposed model is obviously better than that of the baselines. Relative to the current best results, the average improvements obtained on four sub-datasets randomly selected from the Amazon dataset are approximately 3.29%.

Keywords: user reviews; summarization; pre-trained model; rate prediction; recommendation system

Citation: Bai, Y.; Li, Y.; Wang, L. A Joint Summarization and Pre-Trained Model for Review-Based Recommendation. *Information* **2021**, *12*, 223. https://doi.org/10.3390/info12060223

Academic Editor: Ida Mele

Received: 28 April 2021
Accepted: 18 May 2021
Published: 24 May 2021

1. Introduction

With the increasing abundance of products, research on high-quality recommendation systems, especially for the task of rate prediction, has become very important for online e-commerce platforms and users. Most early recommendation systems use collaborative filtering (CF), including user-based collaborative filtering and item-based collaborative filtering. A user-based collaborative filtering method makes recommendations by calculating the similarities between users, while an item-based collaborative filtering method makes recommendations based on the similarities between items. However, CF has its own limitations and drawbacks. First, it has difficulty generating reliable recommendations for users or items with few ratings (the well-known cold-start problem). Another drawback of CF technology is that it does not make full use of the available context information. In other words, context information, such as item attributes [1] or user profiles, is not considered when making recommendations.

Currently, many e-commerce websites not only encourage users to rate products but also encourage users to write product-related reviews. Users can comment on the advantages or disadvantages of the product as well as their experiences with the product in their reviews. User reviews supplement the rating process by providing a wealth of information about the item and the implicit preferences of the users. In addition, these reviews also explain why a user assigned a given rate to the corresponding item [2]. Therefore, to some extent, reviews can help users make purchase decisions, help companies

make marketing decisions and provide interpretability for recommendation systems. As a result, user reviews are being gradually introduced into CF methods to alleviate the above problems.

Intuitively, to make full use of users' reviews, we can infer a user's preferences from all reviews made by her; similarly, reviews of an item describe its outstanding attributes. Inspired by the successful use of deep neural networks on natural language processing (NLP) tasks, some recent works have been devoted to modeling user reviews using deep learning approaches. Common approaches usually concatenate all reviews (user reviews and item reviews) first and then employ neural-network-based methods (e.g., convolutional neural networks (CNNs) [3,4], gated recurrent units (GRUs), and long short-term memory (LSTM) [3,5–9]) to extract a vector representation of the concatenated reviews. However, not all of the reviews are useful for the given recommendation task. To capture the key information in the comments, some models use attention mechanisms to emphasize the key information [10–13].

Commonly, in the above methods, a review document set is regarded as a set of sentences, and all operations are carried out on the sentence set. However, the lengths of review documents are different, and the relationships between sentences are also different. The above methods lose the semantic and global information inside the review text. To this end, we propose modeling user reviews via a Joint Summarization and Pre-Trained Recommendation model (JSPTRec) for the task of rate prediction. This model applies automatic text summarization and compresses the review of a user or an item into a brief summary. In this way, not only the key information but also the relationships between words and sentences in the review are preserved. Then, we use a pre-trained model called "bidirectional encoder representations from transformers" (BERT) [14] to learn the deep semantic representations of the summaries. To capture more fine-grained user preferences or item properties, we use an attention mechanism to distinguish between different review summaries by interacting with user and item vectors. Finally, we try to incorporate the review information of users and items into a neural CF framework [15] to predict the final rating score. To the best of our knowledge, this is the first work to combine automatic summarization and a pre-trained model into a neural recommendation framework used for rate prediction. We compare our method with several competitive baselines on the Amazon dataset, and the experimental results demonstrate that our method is obviously better than other methods. The average improvement is approximately 3.29% over the current best results on the four utilized datasets. We also carry out an ablation study to verify the effectiveness of each part of the JSPTRec model.

2. Methods

In this section, we introduce our recommendation method, which models user reviews via a joint summarization and pre-trained model for rate prediction. The overall model is shown in Figure 1, and it consists of four parts, namely a review summarization layer, a BERT representation layer, an interactive attention layer and a rate prediction layer. Table 1 shows the notations use with the model. A basis of all four parts is that we assume that there exists a K-dimensional latent factor space. Each user or each item is represented as a feature vector in this K-dimensional space, and a user's rating of an item can be calculated by the corresponding feature vectors. We use v_u^{user} to denote the vector for user u and v_i^{item} to denote the vector for item i. For user u, $rw_{u,j}^{user}$ is the j-th review in u's review set $C_u^{user} = \left\{rw_{u,1}^{user}, rw_{u,2}^{user}, \ldots, rw_{u,n}^{user}\right\}$. $e_{u,j}^{item}$ is the corresponding item ID embedding of review $rw_{u,j}^{user}$. Similarly, for an item i, $rw_{i,j}^{item}$ is the j-th review in i's review set $C_i^{item} = \left\{rw_{i,1}^{item}, rw_{i,2}^{item}, \ldots, rw_{i,m}^{item}\right\}$. $e_{i,j}^{user}$ is the corresponding user ID embedding of review $rw_{i,j}^{item}$. For a pair containing user u and item i, we define an affinity score $rate_{u,i}$ that models user u 's preference for item i.

Table 1. Notations.

Notations	Definitions
v_u^{user}	User u's vector
v_i^{item}	Item i's vector
n	The number of reviews for a user
m	The number of reviews for an item
$rw_{u,j}^{user}$	User u's j-th review
C_u^{user}	User u's review set $C_u^{user} = \left\{rw_{u,1}^{user}, rw_{u,2}^{user}, \ldots, rw_{u,n}^{user}\right\}$
$e_{u,j}^{item}$	The corresponding item id embedding of user u's j_{th} review $rw_{u,j}^{user}$
$s_{u,j}^{user}$	User u's j-th review summary
S_u^{user}	User u's summary set $S_u^{user} = \left\{s_{u,1}^{user}, s_{u,2}^{user}, \ldots, s_{u,n}^{user}\right\}$
$rw_{i,j}^{item}$	Item i's j-th review
C_i^{item}	Item i's review set $C_i^{item} = \left\{rw_{i,1}^{item}, rw_{i,2}^{item}, \ldots, rw_{i,m}^{item}\right\}$
$e_{i,j}^{user}$	The corresponding user id embedding of item i's jth review $rw_{i,j}^{item}$
$s_{i,j}^{item}$	Item i's j-th review summary
S_i^{item}	Item i's summary set $S_i^{item} = \left\{s_{i,1}^{item}, s_{i,2}^{item}, \ldots, s_{i,m}^{item}\right\}$
$rate_{u,i}$	User u's rate score for item i

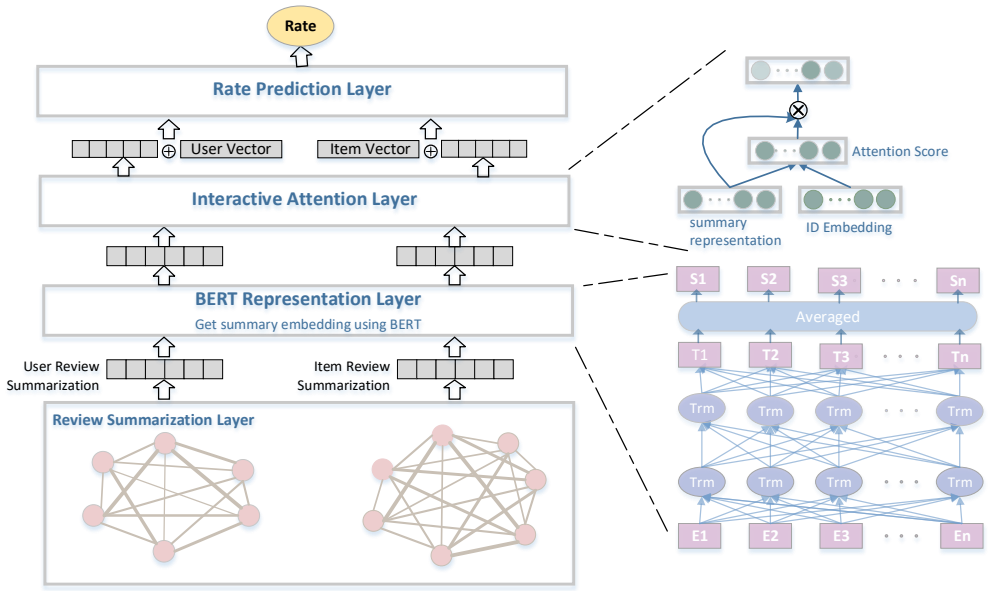

Figure 1. A Joint Summarization and Pre-Training model for Recommendation (JSPTRec).

2.1. Review Summarization Layer

To remove redundant information and retain useful information in the reviews, we used the unsupervised algorithm *TextRank* [16] to extract a summary of each review. *TextRank* is a graph-based ranking algorithm that models a review as a graph $G = (V, E)$. The node set V represents the sentences in the review. E is the set of edges, the weights of which represent the similarities between sentences. The similarity between sentence V_i and sentence V_j can be calculated with the following formula:

$$w_{ij} = Similarity(V_i, V_j) = \frac{\left|\{word_k \mid word_k \in V_i \& word_k \in V_j\}\right|}{log(|V_i|) + log(|V_j|)} \quad (1)$$

where w_{ij} is the weight of the edge between node V_i and node V_j, $word_k$ is a word shared by the sentences, $\left|\{word_k \mid word_k \in V_i \& word_k \in V_j\}\right|$ is the number of words common to

sentence V_i and sentence V_j, $|V_i|$ is the length of sentence V_i, and $|V_j|$ is the length of sentence V_j.

We give each node an initial value that indicates the importance of each sentence and then iteratively update the values of the nodes with the following formula:

$$WS(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} \frac{w_{ij}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j) \quad (2)$$

where $WS(V_i)$ is the value of node V_i, $In(V_i)$ is the set of nodes pointing to V_i, $Out(V_i)$ is the set of nodes to which V_i points, w_{ij} is the weight between node V_i and node V_j, and d is a damping factor between 0 and 1.

Through multiple iterations, the values of the nodes tend to converge, and the values reflect the importance of the nodes. Because these nodes correspond to the sentences in the user review, these values also represent the importance of the sentences in the reviews. Then, we rank the important scores of the sentences and choose the top-n sentences as the summary of the review.

For each review $rw_{u,j}^{user}$, we calculate the importance of each sentence in the $rw_{u,j}^{user}$ through *TextRank* method and then select the most important K sentences as summary $s_{i,j}^{item}$. K can be obtained by $\mu \times |rw_{u,j}^{user}|$, where μ is the proportion of the review summary and $|rw_{u,j}^{user}|$ is the number of sentences in the review. Similarly, for each item review $rw_{i,j}^{item}$, the summary is calculated in the same way.

Therefore, user u's original review set $C_u^{user} = \left\{rw_{u,1}^{user}, rw_{u,2}^{user}, \ldots, rw_{u,n}^{user}\right\}$ and item i's review set $C_i^{item} = \left\{rw_{i,1}^{item}, rw_{i,2}^{item}, \ldots, rw_{i,m}^{item}\right\}$ can be replaced by the corresponding summary sets $S_u^{user} = \left\{s_{u,1}^{user}, s_{u,2}^{user}, \ldots, s_{u,n}^{user}\right\}$ and $S_i^{item} = \left\{s_{i,1}^{item}, s_{i,2}^{item}, \ldots, s_{i,m}^{item}\right\}$, respectively.

2.2. BERT Representation Layer

After obtaining the summaries of the user reviews and item reviews, we use the BERT model [14] to further learn the text representations of the summaries. BERT is an effective pre-trained model that builds upon the transformer architecture. First, it randomly masks 10% to 15% of the words and tries to predict those masked words. Second, BERT takes an input sentence and a candidate sentence and then predicts whether the candidate sentence follows the input sentence. We choose $BERT_{BASE}$ with 12 layers, 768 hidden dimensions, 12 heads, and 110 M parameters as our initial embedding model. The BERT parameters are fine-tuned during the training process of our model. Each summary s is represented as a matrix $\mathbf{R}_s^{W \times E}$, where W is the length of the summary and E is the embedding dimensionality of the words. Then, we perform an average pooling operation over the user's summary and item's summary separately. Hence, each summary is represented as an E-dimensional vector. Finally, the user's summary set $S_u^{user} = \left\{s_{u,1}^{user}, s_{u,2}^{user}, \ldots, s_{u,n}^{user}\right\}$ and the item's summary set $S_i^{item} = \left\{s_{i,1}^{item}, s_{i,2}^{item}, \ldots, s_{i,m}^{item}\right\}$ can be represented as comprehensive summary vectors $S_u^{user} \in \mathbf{R}^{n \times E}$ and $S_i^{item} \in \mathbf{R}^{m \times E}$, respectively.

2.3. Interactive Attention Layer

To focus on the review summaries that are important for predicting user preferences, we use an attention mechanism to capture the interactions between the user and the corresponding item. Given a set of summary representations $S_u^{user} = \left\{s_{u,1}^{user}, s_{u,2}^{user}, \ldots, s_{u,n}^{user}\right\}$ for user u, we can calculate the attention score $\alpha_{u,j}^{user}$ of $s_{u,j}^{user}$ with the following formulas:

$$\alpha_{u,j}^{user*} = ReLU(s_{u,j}^{user} W_1 + e_{u,j}^{item} W_2 + b_1) W_3 + b_2 \quad (3)$$

$$\alpha_{u,j}^{user} = \frac{exp(\alpha_{u,j}^{user*})}{\sum_{k=1}^{n} exp(\alpha_{u,k}^{user*})} \quad (4)$$

where $W_1, W_2 \in R^{E\times E}$, $W_3 \in R^{E\times 1}$, and $b_1 \in R^E, b_2 \in R$ are all trainable parameters; $e^{item}_{u,j}$ is the corresponding item id embedding of user u's review $rw^{user}_{u,j}$; n is the number of reviews provided by a user; and ReLU [17] is a nonlinear activation function:

$$ReLU(x) = \begin{cases} x & if \quad x > 0 \\ 0 & if \quad x \leq 0 \end{cases} \tag{5}$$

For $j = 1, 2, 3, \ldots, n$, we can obtain attention scores $\alpha^{user}_u = \left\{\alpha^{user}_{u,1}, \alpha^{user}_{u,2}, \ldots, \alpha^{user}_{u,n}\right\}$ for the user's summary $S^{user}_u = \left\{s^{user}_{u,1}, s^{user}_{u,2}, \ldots, s^{user}_{u,n}\right\}$.

Similarly, we can calculate the attention scores $\alpha^{item}_i = \left\{\alpha^{item}_{i,1}, \alpha^{item}_{i,2}, \ldots, \alpha^{item}_{i,m}\right\}$ for item i's summary set $S^{item}_i = \left\{s^{item}_{i,1}, s^{item}_{i,2}, \ldots, s^{item}_{i,m}\right\}$ with the following formulas:

$$\alpha^{item*}_{i,j} = ReLU(s^{item}_{i,j} W_1 + e^{user}_{i,j} W_2 + b_1) W_3 + b_2 \tag{6}$$

$$\alpha^{item}_{i,j} = \frac{exp(\alpha^{item*}_{i,j})}{\sum_{k=1}^{m} exp(\alpha^{item*}_{i,k})} \tag{7}$$

where $\alpha^{item}_{i,j}$ is the attention score of item i's j-th summary $s^{item}_{i,j}$, $e^{user}_{i,j}$ is the corresponding user ID embedding of item i's review $rw^{item}_{i,j}$, and m is the number of reviews for an item. For $j = 1, 2, 3, \ldots, m$, we can obtain attention scores $\alpha^{item}_i = \left\{\alpha^{item}_{i,1}, \alpha^{item}_{i,2}, \ldots, \alpha^{item}_{i,m}\right\}$.

Subsequently, the final review representation vector of user reviews and item reviews can be obtained by a weighted summation of the summary representations of all reviews with the following formulas:

$$A^{user}_u = \alpha^{userT}_u S^{user}_u \tag{8}$$

$$A^{item}_i = \alpha^{itemT}_i S^{item}_i \tag{9}$$

2.4. Rate Prediction Layer

To focus on the effective information in the reviews used for recommendation, we incorporate the summaries of user u and item i to obtain review representation vectors A^{user}_u and A^{item}_i, respectively. We then obtain the feature vectors of the users and items through their review representation vectors. F_u , $F_i \in R^D$ are the final feature vectors of user u and item i with the following formulas:

$$F_u = v^{user}_u + A^{user}_u \cdot W_u + b_u \tag{10}$$

$$F_i = v^{item}_i + A^{item}_i \cdot W_i + b_i \tag{11}$$

where $W_u, W_i \in R^{E\times D}$ are trainable weighted parameters, D is the dimensionality of the feature vector, and $b_u \in R^D$ and $b_i \in R^D$ are the bias vectors of the user and item that can record long-term information.

Finally, for a pair containing user u and item i, the affinity score $rate_{u,i}$ can be viewed as user u's preference for item i with the following formula:

$$\hat{rate}_{u,i} = (F_u \odot F_i) \cdot W + b \tag{12}$$

where $W \in R^{D\times 1}$ and $b \in R$ are trainable parameters and b is the bias of the rating score.

We use the mean squared error (*MSE*) as the loss function to train our model, and we optimize the model by minimizing the *MSE* between the output score from our model $\hat{rate}_{u,i}$ and the real score $rate_{u,i}$ with the following formula:

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (rate_{u,i} - \hat{rate}_{u,i})^2 \tag{13}$$

3. Results and Discussion

In this section, we empirically evaluate the various components of our proposed JSPTRec model for rate prediction. We conduct experiments to answer the following research questions: (i) How much can user reviews help rate prediction compared with CF baselines? (ii) Does our method perform better than other baseline methods that also use a hybrid CF and review-based recommendation approach? (iii) Can a summarization and pre-trained model help accomplish recommendation tasks, and if so, which part is most useful?

3.1. Dataset and Evaluation Metric

We conducted experiments on four different datasets from Amazon Review Data (https://nijianmo.github.io/amazon/index.html) (accessed on 2 April 2019). Table 2 shows the numbers of users, items, and reviews in each dataset. The Amazon Review dataset includes reviews (ratings, text, helpfulness votes), product metadata (descriptions, category information, prices, brands, and image features), and links (also viewed/also bought graphs). For each dataset, we randomly select 80% of the user–item pairs as the training set, 10% as the validation set, and 10% as the testing set. We use only the reviews in the training set to learn representations for the users and items and do not use the reviews in the validation and testing sets.

Table 2. Dataset statistics.

Dataset	Users	Items	Reviews
Automotive	2928	1835	20,468
Digital Music	2986	551	9999
Musical Instruments	1429	900	10,255
Patio Lawn and Garden	1686	962	13,254

In our experiments, we adopt the widely used MSE to evaluate the performances of the compared algorithms.

3.2. Compared Methods

We compare our model with several competitive baselines, including CF-based methods and deep learning-based methods, by using reviews. To perform the experiments, we use an open source code on github (https://github.com/JieniChen/Recommender-System) (accessed on 21 April 2020) for PMF and NMF. For other baselines, we use the codes provided by the authors, respectively.

- Probabilistic matrix factorization (PMF (https://github.com/JieniChen/Recommender-System) (accessed on 21 April 2020)) [18]: PMF is a widely used rating-based CF method. PMF assumes that the elements in the scoring matrix are determined by the inner product of the user's potential preference vector and the item's potential attribute vector.
- Nonnegative matrix factorization (NMF (https://github.com/JieniChen/Recommender-System) (accessed on 21 April 2020)) [19]: NMF is also a rating-based CF method. It assumes that the decomposed matrix should satisfy nonnegativity constraints. NMF can decompose a nonnegative matrix into two nonnegative matrices.
- Hidden factors and hidden topics (HFT (http://cseweb.ucsd.edu/jmcauley/code/code_RecSys13.tar.gz) (accessed on 9 February 2021)) [20]: HFT models the given ratings using a matrix factorization model with an exponential transformation function to link the stochastic topic distribution obtained from modeling the review text and the latent vector obtained from modeling the ratings. It assumes that the topic distribution of each review is produced on either user factors or item factors. In this way, HFT can provide an interpretation of each latent factor because factors and topics are located in the same space.

- Deep Cooperative Neural Networks (DeepCoNN (https://github.com/richdewey/DeepCoNN) (accessed on 16 March 2019)) [6]: DeepCoNN uses a CNN to model user reviews to learn the underlying user behaviors and product attributes. DeepCoNN constructs two parallel networks and connects them with a shared additional layer so that the interaction between the user and the product can be used to predict the final score.
- Multipointer coattention network (MPCN (https://github.com/vanzytay/kdd2018_mpcn) (accessed on 17 March 2019)) [10]: The MPCN is based on the idea that a few reviews are important and that the importance depends dynamically on the current target. To extract important reviews, the MPCN contains a review-by-review pointer-based learning scheme that matches reviews in a word-by-word fashion. The pointer mechanism used in the MPCN is essentially coattentive and can learn the dependencies between users and items.
- Dual attention-based model (D-Attn (https://hub.fastgit.org/seongjunyun/CNN-with-Dual-Local-and-Global-Attention) (accessed on 17 March 2019)) [2]: D-Attn uses aggregated review texts from a user and an item to learn the embeddings of the user and the item. D-Attn applies CNNs with dual (local and global) attention mechanisms. Local attention is used to capture a user's preferences or an item's properties. Global attention helps the CNNs focus on the semantic information of the review text.
- Neural attentional regression model with review-level explanations (NARRE (https://github.com/chenchongthu/narre) (accessed on 6 April 2019)) [21]: The NARRE proposes an attention mechanism to explore the usefulness of different reviews. The weights of reviews are learned by an attention mechanism in a distant supervised manner. Moreover, the NARRE learns the latent features of users and items using two parallel neural networks.

3.3. Experimental Settings

For our JSPTRec model, the dimensionality of the user and item vectors and their ID embedding vectors is set to 32. The learning rate is set to 0.001. The proportion of the summary extracted from the review is set to 0.6. For all the baselines, we use the same settings as those in their original papers. For PMF and NMF, the number of factors is set to 100. For HFT, the latent dimensions and number of topics are both determined by the parameter K. We set $K = 10$, which is the same as that in the original paper. For DeepCoNN, the number of convolutional kernels is set to 100, and the window size is 3. D-Attn uses 200 filters and a window size of 5 for local attention, and it uses 100 filters and window sizes of [2, 3, 4] for global attention. The MPCN uses three pointers and 300 hidden dimensions to infer the affinity matrix.

3.4. Experimental Results

In Table 3, we compare the results of our method and the baseline methods on four different datasets. From the experimental results in Table 3, we can draw the following conclusions:

(i) HFT, D-Attn, DeepCoNN, the MPCN, and the NARRE generally perform better than PMF and NMF because the review-based methods benefit from the introduction of user reviews. This indicates that user reviews are helpful for completing recommendation tasks.

(ii) D-Attn, DeepCoNN, the MPCN, and the NARRE outperform HFT, indicating that the deep-learning-based methods are more effective than CF-based methods in terms of modeling user reviews and understanding the semantic information in text.

(iii) By selecting or weighting user reviews, D-Attn, the MPCN, and the NARRE outperform DeepCoNN, which suggests that different reviews exhibit different importance levels for modeling users and items in rate prediction tasks.

(iv) The proposed JSPTRec model outperforms all the baseline methods. This shows that the recommendation method based on summarization and a pre-trained model is

effective, as this approach can retain important review information and obtain the best recommendation results.

Table 3. Experimental results on four subdatasets from Amazon review data.

Dataset	Automotive	Digital Music	Musical Instruments	Patio Lawn
PMF	1.1515	0.9769	1.1251	1.7378
NMF	1.0878	0.7699	1.0058	1.2330
HFT	1.1415	0.7744	1.0104	1.1406
DeepCoNN	0.8681	0.9830	0.7146	1.1193
D-Attn	0.8615	0.8600	0.7493	1.0958
MPCN	0.9400	0.7433	0.7314	1.0835
NARRE	0.8225	0.8396	0.6829	1.1055
JSPTRec	0.8191	0.7130	0.6856	1.0361

3.5. Parameter Sensitivity Analysis

We would like to analyze how sensitive the performance of our model is with regard to the parameters on the Musical Instruments dataset. First, we varied the dimensionality of the user and item vectors while fixing the other parameters. Figure 2 shows the performances of PMF, NMF, DeepCoNN, the NARRE, and our model. PMF is greatly influenced by dimensionality, and the accuracy of its predictions increases significantly with increasing dimensionality. NMF, DeepCoNN, and the NARRE all have stable performances with different numbers of dimensions, and our model achieves the best performance with all dimensionality settings.

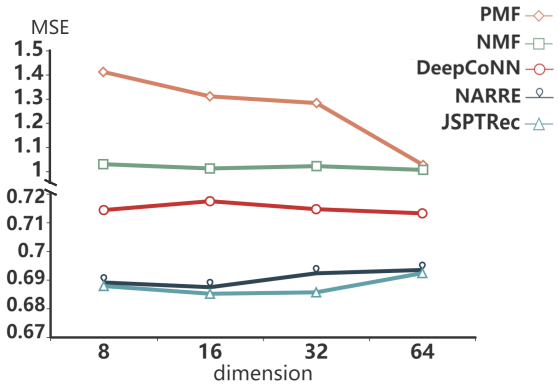

Figure 2. MSEs on the Musical Instruments dataset with different numbers of dimensions for the user and item vectors.

Then, we tested our model with different proportions of the summary extracted from the review. We set the proportion μ to 0.2, 0.4, 0.6, 0.8, and 1. In Figure 3, with the increase in μ, more information in the review is retained. When $\mu = 0.2$ or 0.4, the recommendation effect is poor because too little review text is extracted, resulting in the loss of some of the valid information. We find that when $\mu = 0.6$, the best results can be achieved. When $\mu = 0.8$ or 1, the recommendation effect is slightly worse than that when $\mu = 0.6$, which means that we can obtain almost all the useful semantic information from reviews by using only 60% of the review text, and too much text introduces noise.

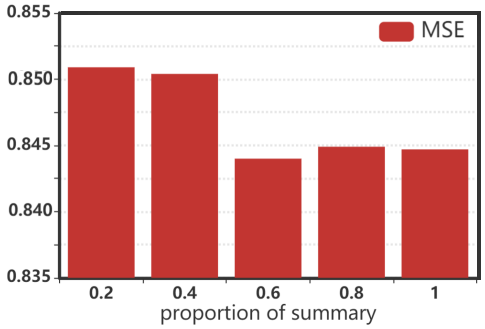

Figure 3. MSEs with different proportions of the review summary.

3.6. Ablation Study

To test the effectiveness of each part of our model, we also conducted ablation experiments. JSPTRec-BERT, JSPTRec-TR, and JSPTRec-ATT are three weaker variations of our complete model. JSPTRec-BERT represents our model without the pre-trained model (BERT), in which the word embeddings are initialized by Glove. JSPTRec-TR represents our model without the summarization model (TextRank). JSPTRec-ATT represents our model without an interactive attention layer. In JSPTRec-TR, all the user reviews from a given user or item are concatenated into a long document as the input. From Table 4, we can find that JSPTRec outperforms JSPTRec-BERT, which demonstrates that the pre-trained model is effective in learning deep user preferences and item properties from user review texts. Furthermore, our model is even better than the model without summarization, JSPTRec-TR (using all review texts), indicating that the summarization layer can retain the "key" information from the review text and reduce the calculations required of the model. Finally, JSPTRec obtains superior results to those of JSPTRec-ATT, which shows the effectiveness of the interactive attention mechanism.

Table 4. Ablation Study.

Dataset	Automotive	Digital Music	Musical Instruments	Patio Lawn
JSPTRec-BERT	0.7845	0.6632	0.8493	1.0402
JSPTRec-TR	0.7803	0.6765	0.8447	1.0369
JSPTRec-ATT	0.8490	0.6811	0.8493	1.0539
JSPTRec	0.7771	0.6314	0.8440	0.9967

4. Related Work

With the increasing amount of network information, recommendation systems have become widely used [22]. In this section, we present three lines of work that are related to our task, namely, CF-based recommendation, review summarization, and deep-learning-based review modeling.

4.1. Collaborative Filtering Based Recommendation

CF [23] uses the aggregated behaviors/tastes of a large number of users to suggest relevant items to specific users. Recommendations generated by CF are based solely on user–user and/or item–item similarities, which are popular and widely deployed by Internet companies such as Amazon [24], Google News [25], and others. In addition to CF based on users and items, another kind of method exists: model-based CF. The main idea of matrix factorization is to construct an implicit semantic model; that is, by decomposing the sorted and extracted "user item" scoring matrix, a user latent vector matrix and an item latent vector matrix can be obtained. There are many matrix factorization models, such as the latent factor model (LMF), singular value decomposition (SVD), and PMF. PMF [18] is widely used because it scales linearly with the number of observations and performs well on very sparse and imbalanced datasets. To improve the interpretability of the model,

NMF [19] imposes a nonnegativity constraint on the two decomposed small matrices on the basis of SVD.

Cold start is a common problem in CF-based recommendation systems. Many scholars have tried to alleviate this problem by introducing various external information. Dhelim et al. [26] proposed a product recommendation system based on user interest mining and metapath discovery to alleviate the cold start problem. In addition, users' social relations also contain rich user characteristics. Khelloufi et al. [27] took advantage of the social relations between devices to select a suitable service that fits the requirements of the applications and devices, based on the observation that having a given personality type does not necessarily mean that you are compatible with people sharing the same personality type. Ning et al. [28] designed a friend recommendation system based on the big-five personality traits model and hybrid filtering, in which the friend recommended process is based on personality traits and users' harmony rating.

Using user reviews to alleviate the cold-start problem in CF has attracted extensive attention in recent years. Wang and Blei [29] first combined the merits of traditional CF and probabilistic topic modeling. The clickthrough rate (CTR) model integrates PMF and latent Dirichlet allocation (LDA) into the same probability framework in a tightly coupled way. HFT [20] models user reviews with matrix factorization and assumes that the topic distribution of each review is produced by the latent factors of the corresponding item. King [30] proposed a unified model called "ratings meet reviews" (RMR) that combines content-based filtering with CF, harnessing the information of both ratings and reviews. RMR applies topic modeling techniques to the review text and aligns the topics with rating dimensions to improve prediction accuracy.

In recent years, CF has been combined with deep learning models. Most matrix factorization methods apply an inner product to the latent features of users and items. Salakhutdinov et al. [31] demonstrated that restricted Boltzmann machines (RBMs) can be applied to rate prediction tasks and slightly outperform carefully tuned SVD models. By replacing the inner product with a neural architecture that can learn an arbitrary function from data, He et al. [15] developed a general framework called neural collaborative filtering (NCF) and proposed leveraging a multilayer perceptron to learn nonlinear user–item interaction functions.

4.2. Review Summarization

With the deepening and increasing number of product reviews, it is a growing challenge for customers and product manufacturers to gain a comprehensive understanding of their contents. Automatic summarization of reviews aims to mine and summarize all the customer reviews of a product, which can capture important information. It is a key step for review document understanding and sentiment analysis [32,33]. Shimada et al. [34] proposed a method for generating a summary that contains sentiment information and objective information of a product. The authors use three features: ratings of aspects, the value, and the number of mentions with a similar topic to generate a more appropriate summary. Due to the importance of the product feature and opinion extraction to review summarization, Nyaung and Thein [35] refer the task of review summarization to relating the opinion words with respect to a certain feature. Mabrouk et al. [36] proposed a methodology to summarize aspects and spot opinions regarding them using a combination of template information with customer reviews in two main phases. Recently, some deep neural models have been used in review summarization. In order to achieve generative review summarization, a neural attention network model with sequence-to-sequence learning was conducted [37]. By focusing on the feature of review summarization samples, the local attention mechanism is improved that has more attention weights on the start of the source sentence. Then, each word of the summary is generated through the end-to-end model. Xu et al. [38] proposed a neural review-level attention model to effectively learn user preference embedding and product characteristic embedding from their history reviews. Then, they designed a personalized decoder to generate the personalized summary, which

utilizes the representations of the user and the product to calculate saliency scores for words in the input review to guide the summary-generation process. Finally, a multi-task framework was used to joint optimize the summary generation and rating prediction.

4.3. Deep Learning Based Review Modeling

Users often post reviews on the Internet, and these reviews contain rich semantic information about users and items. Recently, some works have employed deep learning algorithms to model auxiliary review information, such as the textual descriptions of items and preferences of users.

To capture multiangle features or multilevel features, some methods apply CNNs [4] to model user reviews. Seo et al. [2] proposed a CNN-based recommendation model with local and global attention. Kim et al. [39] combined CNNs with PMF to better capture contextual information. However, important semantic features may be contained in text segments of different granularities. Wang et al. [3] designed a hierarchical and fine-grained CNN-based recommendation model that can obtain multilevel user/item representations and match them separately. In addition to CNN models, the recurrent neural network (RNN) model and its variants (GRUs and LSTM) have also been adopted to extract much semantic information from user reviews [40–45]. Li et al. [45] used gated RNNs to learn user and item latent representations from reviews. They designed a sequence decoding model based on a gated RNN called a GRU. This model not only predicts ratings but also generates abstractive tips based only on user latent factors and item latent factors [45].

To further select important information from review texts, attention mechanisms have been used for user review modeling [10,46–50]. Attention mechanisms can focus on important information or capture the correlations between users and items. Chin et al. [46] merged all reviews provided by a given user into a long document to extract aspect-level representations of users and items. Then, they used a coattention mechanism to build the correlation matrix between the users and items. To capture the relationship between reviews and a target item, Tay et al. [10] applied an attention mechanism at both the review level and word level to dynamically select important reviews for the target item. Similarly to Tay et al. [10], Liu et al. [47] also used multilayer attention. They utilized local and mutual attention on top of CNNs to jointly learn the features of reviews. Zhao et al. [48] used explicit behavior factors, such as retweeting and mentioning, to understand users and utilized an attention mechanism that could automatically learn the weights of different factors. However, existing techniques mainly extract the latent representations of users and items in an independent and static manner. Wu et al. [51] proposed a novel context-aware user and item representation learning model that uses two separate learning components to exploit review data and interaction data: review-based feature learning and interaction-based feature learning, respectively.

Some models also use additional information to supplement review-based recommendations. For example, Ye et al. [52] used not only reviews but also product images. They presented a novel collaborative neural model for rating prediction by jointly utilizing user reviews and product images. They coupled the processes of rating prediction and review generation via a deep neural network and generated review content using an LSTM-based model. Probability-based methods are also used in rating prediction. Lei et al. [53] used LDA, which is a Bayesian model, to model the relationships between reviews, topics, and words.

In contrast, pre-trained language models such as Elmo [54], the generative pre-trained transformer (GPT) [55,56], and BERT [14] have shown good performance on many NLP tasks. The existing pre-trained language models are mainly based on RNNs [54,57,58] and transformers [14,55,56]. Among these, BERT (Bidirectional Encoder Representations from Transformers [14]) is a very effective pre-trained model that can obtain bidirectional representations of context.

Motivated by the above successes, we propose modeling user reviews via a joint summarization and pre-trained model for the task of rate prediction. We perform au-

tomatic text summarization to compress all reviews into a brief summary that not only extracts the key information but also preserves the relationships between the words and sentences in the review. Then, a pre-trained model (BERT [14]) is used to learn the deep semantic representations of the summaries, and interactive attention is used to focus on important information and produce high-quality summary representations. Finally, we try to incorporate the review summary representations of users and items into a neural CF framework [15] to predict the rating score. To the best of our knowledge, this is the first work that combines automatic summarization and a pre-trained model into a neural recommendation framework for the task of rate prediction.

5. Conclusions

In this paper, we proposed a joint summarization and pre-trained recommendation model called JSPTRec for review-based recommendation. The model benefits from automatic summary extraction, a pre-trained model, and interactive attention mechanisms. We designed experiments to evaluate our model against several state-of-the-art models. Via a comparison with CF-based methods, we found that user reviews were significantly helpful, indicating that it is important to introduce user review texts for rate prediction. Second, we found it beneficial to perform summarization to capture the important information from a large number of reviews. Third, we found that compared with other deep-learning-based methods, the pre-trained model BERT can learn better semantic representations of reviews for users and items. Finally, by using interactive user and item attention mechanisms, the recommendation performance of our model can be further improved.

Author Contributions: Conceptualization, Y.B., Y.L. and L.W.; methodology, Y.L.; software, Y.B.; validation, Y.B. and Y.L.; formal analysis, Y.L.; investigation, Y.B. and L.W.; resources, Y.L.; data curation, Y.L.; writing—original draft preparation, Y.B.; writing—review and editing, Y.L.; visualization, Y.B.; supervision, Y.L.; project administration, Y.L.; funding acquisition, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: We thank the anonymous reviewers for their constructive suggestions. This work was supported by the National Natural Science Foundation of China [61806049] and the Heilongjiang Postdoctoral Science Foundation [LBH-Z20104] and the National Undergraduate Training Programs for Innovation and Entrepreneurship via grant number [202010225020].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The code and data can be found at https://github.com/catly/SumBERT-JSPTRec (accessed date 20 May 2021).

Acknowledgments: We thank the anonymous reviewers for their constructive suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shakhovska, N.; Fedushko, S.; Shvorob, I.; Syerov, Y. Development of Mobile System for Medical Recommendations. *Procedia Comput. Sci.* **2019**, *155*, 43–50. [CrossRef]
2. Seo, S.; Huang, J.; Yang, H.; Liu, Y. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In Proceedings of the Eleventh ACM Conference on Recommender Systems, Como, Italy, 27–31 August 2017; pp. 297–305.
3. Wang, H.; Wu, F.; Liu, Z.; Xie, X. Fine-grained Interest Matching for Neural News Recommendation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 5–10 July 2020; pp. 836–845.
4. Bouvrie, J. Notes on Convolutional Neural Networks; Neural Nets. MIT CBCL Tech Report. 2006. Available online: http://cogprints.org/5869/1/cnn_tutorial.pdf (accessed on 25 April 2020).
5. Gong, Y.; Zhang, Q. Hashtag recommendation using attention-based convolutional neural network. In Proceedings of the IJCAI, New York, NY, USA, 9–15 July 2016; pp. 2782–2788.
6. Zheng, L.; Noroozi, V.; Yu, P.S. Joint deep modeling of users and items using reviews for recommendation. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, Cambridge, UK, 6–10 February 2017; pp. 425–434.

7. Wu, Y.; DuBois, C.; Zheng, A.X.; Ester, M. Collaborative Denoising Auto-Encoders for Top-N Recommender Systems. In Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, San Francisco, CA, USA, 22–25 February 2016; pp. 153–162.
8. Jing, H.; Smola, A.J. Neural survival recommender. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, Cambridge, UK, 6–10 February 2017; pp. 515–524.
9. Bansal, T.; Belanger, D.; McCallum, A. Ask the gru: Multi-task learning for deep text recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016; pp. 107–114.
10. Tay, Y.; Luu, A.T.; Hui, S.C. Multi-pointer co-attention networks for recommendation. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 2309–2318.
11. Luo, A.; Zhao, P.; Liu, Y.; Zhuang, F.; Sheng, V.S. Collaborative Self-Attention Network for Session-based Recommendation. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence and Seventeenth Pacific Rim International Conference on Artificial Intelligence IJCAI-PRICAI-20, Yokohama, Japan, 11–17 July 2020.
12. Dong, X.; Ni, J.; Cheng, W.; Chen, Z.; Melo, G.D. Asymmetrical Hierarchical Networks with Attentive Interactions for Interpretable Review-Based Recommendation. *Proc. Aaai Conf. Artif. Intell.* **2020**, *34*, 7667–7674. [CrossRef]
13. Zhou, J.P.; Cheng, Z.; Pérez, F.; Volkovs, M. TAFA: Two-headed attention fused autoencoder for context-aware recommendations. In Proceedings of the Fourteenth ACM Conference on Recommender Systems, Rio de Janeiro, Brazil, 22–26 September 2020; pp. 338–347.
14. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
15. He, X.; Liao, L.; Zhang, H.; Nie, L.; Chua, T.S. Neural Collaborative Filtering. In Proceedings of the 26th International Conference on World Wide Web, Perth Australia, 3–7 April 2017. pp. 173–182
16. Mihalcea, R.; Tarau, P. Textrank: Bringing order into text. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 25–26 July 2004; pp. 404–411.
17. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the ICML, Haifa, Israel, 21–24 June 2010.
18. Mnih, A.; Salakhutdinov, R.R. Probabilistic Matrix Factorization. *Adv. Neural Inf. Process. Syst.* **2007**, *20*, 1257–1264.
19. Lee, D.D.; Seung, H.S. Algorithms for Non-Negative Matrix Factorization. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Vancouver, BC, Canada, 3–8 December 2001; pp. 556–562.
20. McAuley, J.; Leskovec, J. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. In Proceedings of the 7th ACM conference on Recommender systems, Hong Kong, China, 12–16 October 2013; pp. 165–172.
21. Chen, C.; Zhang, M.; Liu, Y.; Ma, S. Neural attentional rating regression with review-level explanations. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 1583–1592.
22. Zhang, S.; Yao, L.; Sun, A.; Tay, Y. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv. (CSUR)* **2019**, *52*, 1–38. [CrossRef]
23. Balabanović, M.; Shoham, Y. Fab: Content-based, collaborative recommendation. *Commun. ACM* 1997, *40*, 66–72. [CrossRef]
24. Linden, G.; Smith, B.; York, J. Amazon.com recommendations: Item-to-item collaborative filtering. *Internet Comput. IEEE* **2003**, *7*, 76–80. [CrossRef]
25. Das, A.S.; Datar, M.; Garg, A.; Rajaram, S. Google news personalization: Scal-able online collaborative filtering. In In Proceedings of the 16th International Conference on World Wide Web, Banff, AB, Canada, 8–12 May 2007; pp. 271–280.
26. Dhelim, S.; Ning, H.; Aung, N.; Huang, R.; Ma, J. Personality-Aware Product Recommendation System Based on User Interests Mining and Metapath Discovery. *IEEE Trans. Comput. Soc. Syst.* **2020**. [CrossRef]
27. Khelloufi, A.; Ning, H.; Dhelim, S.; Qiu, T.; Atzori, L. A Social Relationships Based Service Recommendation System For SIoT Devices. *IEEE Internet Things J.* **2020**. [CrossRef]
28. Ning, H.; Dhelim, S.; Aung, N. PersoNet: Friend Recommendation System Based on Big-Five Personality Traits and Hybrid Filtering. *IEEE Trans. Comput. Soc. Syst.* **2019**, *6*, 394–402. [CrossRef]
29. Wang, C.; Blei, D.M. Collaborative Topic Modeling for Recommending Scientific Articles. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 21–24 August 2011.
30. King, G.L.L. Ratings Meet Reviews, a Combined Approach to Recommend. In Proceedings of the RecSys '14, Silicon Valley, CA, USA, 6–10 October 2014; pp. 105–112.
31. Salakhutdinov, R.; Mnih, A.; Hinton, G. Restricted Boltzmann machines for collaborative filtering. In Proceedings of the 24th International Conference on Machine Learning, Corvallis, OR, USA, 20–24 June 2007; pp. 791–798.
32. Wu, P.; Li, X.; Shen, S.; He, D. Social media opinion summarization using emotion cognition and convolutional neural networks. *Int. J. Inf. Manag.* **2020**, *51*, 101978. [CrossRef]
33. Al-Natour, S.; Turetken, O. A comparative assessment of sentiment analysis and star ratings for consumer reviews—ScienceDirect. *Int. J. Inf. Manag.* **2020**, *54*, 102132. [CrossRef]
34. Shimada, K.; Tadano, R.; Endo, T. Multi-aspects review summarization with objective information. *Procedia Soc. Behav. Sci.* **2011**, *27*, 140–149. [CrossRef]
35. Nyaung, D.E.; Thein, T.L.L. Feature-Based Summarizing and Ranking from Customer Reviews. *Int. J. Eng.* **2015**, *9*, 734–739.

36. Mabrouk, A.; Redondo, R.; Kayed, M. SEOpinion: Summarization and Exploration of Opinion from E-Commerce Websites. *Sensors* **2021**, *21*, 636. [CrossRef]
37. Su, F.; Wang, X.; Zhang, Z. Review Summarization Generation Based on Attention Mechanism. *J. Beijing Univ. Posts Telecommun.* **2018**, *41*, 7.
38. Xu, H.; Liu, H.; Zhang, W.; Jiao, P.; Wang, W. Rating-boosted abstractive review summarization with neural personalized generation. *Knowl.-Based Syst.* **2021**, *218*, 106858. [CrossRef]
39. Kim, D.; Park, C.; Oh, J.; Lee, S.; Yu, H. Convolutional Matrix Factorization for Document Context-Aware Recommendation. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016.
40. Ma, D.; Li, S.; Zhang, X.; Wang, H. Interactive attention networks for aspect-level sentiment classification. *arXiv* **2017**, arXiv:1709.00893.
41. Cheng, J.; Dong, L.; Lapata, M. Long short-term memory-networks for machine reading. *arXiv* **2016**, arXiv:1601.06733.
42. Li, Y.; Liu, T.; Jiang, J.; Zhang, L. Hashtag recommendation with topical attention-based LSTM. In Proceedings of the COLING 2016, 26th International Conference on Computational Linguistics, Osaka, Japan, 11–16 December 2016; pp. 943–952.
43. Liu, Q.; Zhang, H.; Zeng, Y.; Huang, Z.; Wu, Z. Content attention model for aspect based sentiment analysis. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 1023–1032.
44. Manotumruksa, J.; Macdonald, C.; Ounis, I. A contextual attention recurrent architecture for context-aware venue recommendation. In Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018; pp. 555–564.
45. Li, P.; Wang, Z.; Ren, Z.; Bing, L.; Lam, W. Neural rating regression with abstractive tips generation for recommendation. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017; pp. 345–354.
46. Chin, J.Y.; Zhao, K.; Joty, S.; Cong, G. ANR: Aspect-based neural recommender. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Turin, Italy, 22–26 October 2018; pp. 147–156.
47. Liu, D.; Li, J.; Du, B.; Chang, J.; Gao, R. 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, Anchorage, AK, USA, 4–8 August 2019; pp. 344–352.
48. Zhao, G.; Lei, X.; Qian, X.; Mei, T. Exploring users' internal influence from reviews for social recommendation. *IEEE Trans. Multimed.* **2018**, *21*, 771–781. [CrossRef]
49. Cheng, Z.; Ding, Y.; He, X.; Zhu, L.; Song, X.; Kankanhalli, M.S. Aˆ 3NCF: An Adaptive Aspect Attention Model for Rating Prediction. In Proceedings of the IJCAI, Stockholm, Sweden, 13–19 July 2018; pp. 3748–3754.
50. Mei, L.; Ren, P.; Chen, Z.; Nie, L.; Ma, J.; Nie, J.Y. An attentive interaction network for context-aware recommendations. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Turin, Italy, 22–26 October 2018; pp. 157–166.
51. Wu, L.; Quan, C.; Li, C.; Wang, Q.; Zheng, B.; Luo, X. A context-aware user-item representation learning for item recommendation. *ACM Trans. Inf. Syst. (TOIS)* **2019**, *37*, 1–29. [CrossRef]
52. Ye, W.; Zhang, Y.; Zhao, W.X.; Chen, X.; Qin, Z. A collaborative neural model for rating prediction by leveraging user reviews and product images. In *Asia Information Retrieval Symposium*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 99–111.
53. Lei, X.; Qian, X.; Zhao, G. Rating prediction based on social sentiment from textual reviews. *IEEE Trans. Multimed.* **2016**, *18*, 1910–1921. [CrossRef]
54. Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Zettlemoyer, L. Deep Contextualized Word Representations. *arXiv* **2018**, arXiv:1802.05365.
55. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. 2018. Available online: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf (accessed on 15 May 2020).
56. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
57. Howard, J.; Ruder, S. Universal Language Model Fine-tuning for Text Classification. *arXiv* **2018**, arXiv:1801.06146.
58. Chronopoulou, A.; Baziotis, C.; Potamianos, A. An Embarrassingly Simple Approach for Transfer Learning from Pretrained Language Models. In Proceedings of the 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), Minneapolis, MN, USA, 2–7 June 2019; pp. 2089–2095.