Information Sciences 567 (2021) 185–200

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

TAERT: Triple-Attentional Explainable Recommendation with Temporal Convolutional Network

Siyuan Guo [a], Ying Wang [a], Hao Yuan [a], Zeyu Huang [a], Jianwei Chen [a], Xin Wang [b,*]

[a] College of Computer Science and Technology, Jilin University, Changchun, Jilin, China
[b] College of Artificial Intelligence, Jilin University, Changchun, Jilin, China

ARTICLE INFO

Article history:
Received 3 May 2020
Received in revised form 29 December 2020
Accepted 12 March 2021
Available online 18 March 2021

Keywords:
Recommender system
Explainable recommendation
Triple attention networks
Temporal Convolutional Network
Rating prediction

ABSTRACT

Explainable Recommendation aims at not only providing the recommended items to users, but also enabling users to be aware of why these items are recommended. To better understand the recommended results, textual reviews have been playing an increasingly important role in the recommender systems. However, how to learn the latent representation of user preferences and item features, and how to model the interactions between them effectively via specific aspects in the reviews are two crucial problems in the explainable recommendation. To this end, we propose a novel Triple-Attentional Explainable Recommendation with Temporal Convolutional Network, named TAERT, which is to jointly generate recommendation results and explanations. Specifically, we first explore a feature learning method based on Temporal Convolutional Network (TCN) to derive word-aware and review-aware vector representations. Then, we introduce three levels of attention networks to model word contribution, review usefulness and importance of latent factors, respectively. Finally, the predicted rating is inferred by the factor-level attention based prediction layer. Furthermore, the attention mechanism is also conducive to identifying the representative item reviews and highlighting the informative words to generate explanations. Compared with the state-of-the-art methods, comprehensive experiments on six real-world datasets are conducted to verify the effectiveness on both recommendation and explanation.

1. Introduction

To alleviate the problem of information overload, recommender systems are widely used in the online platforms, such as Amazon, Netflix and Yelp. However, a large number of recommendation models are still a black-box which do not provide explanations to the user or only gives some simple explanations like "Popular products inspired by this item". An example of Amazon is shown in Fig. 1. Actually, an ideal recommender system should focus on explanation, form a virtuous circle through user feedback, and constantly improve the performance. Explainable recommender systems not only unveil the recommendation process, but also help improve the effectiveness, persuasiveness and satisfaction of recommendation.

Collaborative Filtering (CF) [23] is one of the most popular methods among the recommendation techniques, which mainly utilizes the explicit or implicit feedback to model the interactions between users and items. Among the CF tech-

* Corresponding author.
E-mail addresses: guosy2117@mails.jlu.edu.cn (S. Guo), wangying2010@jlu.edu.cn (Y. Wang), yuanhao2117@mails.jlu.edu.cn (H. Yuan), huangzy2117@mails.jlu.edu.cn (Z. Huang), chenjw2117@ mails.jlu.edu.cn (J. Chen), xinwang@jlu.edu.cn (X. Wang).

https://doi.org/10.1016/j.ins.2021.03.034
0020-0255/

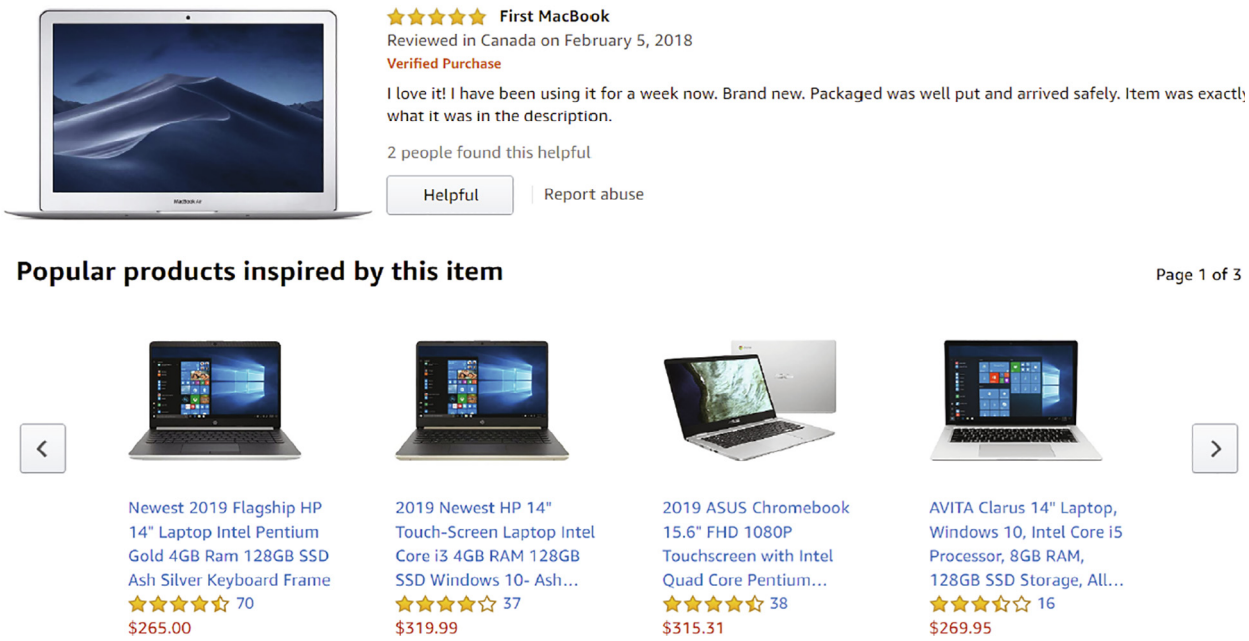

Fig. 1. An example of recommendation in Amazon. The upper half part shows the rating and commenting services while the lower part illustrates the explanations given by Amazon when recommending items to the user.

niques, Matrix Factorization (MF) [30,21] has achieved great success in learning user preferences and item features to predict ratings. The MF is to decompose the user-item rating matrix into two matrices for users' and items' vector representations respectively. However, for those methods only relying on the rating matrix, its recommendation performance tends to decrease when the rating matrix is sparse. What's more, the cold-start issues weaken the performance significantly. To solve these problems, many existing models [28,46,22] consider the textual reviews as the contextual information of user-item interactions and leverage them to improve the recommendation performance.

For the review-based recommender systems, CNN-based feature learning methods are widely applied to learn user preferences and item features from the textual reviews for the supplement of the user-item interactions to enhance their expressive power. However, when processing textual reviews, CNN can only capture neighbor feature interactions and ignore many useful global feature interactions. Furthermore, in terms of natural language, the word sequence is an essential characteristic but usually cannot be well captured by CNN-based feature learning methods. Therefore, in terms of textual reviews in recommender system, it is imperative to construct a feature learning model that fully considers global interaction features and sequence characteristics.

In terms of explainable recommender systems, many existing models exploit the textual reviews to generate readable explanations. Since low-quality reviews may introduce noises, it is particularly important to identify the reviews with the detailed item information. These reviews with high review usefulness [3] are conducive to enhancing the performance of recommendation and providing understandable explanations. Meanwhile, some existing models exploit reviews to identify the aspect information for recommendation explanations because different users tend to express their opinions on different aspects [6,7]. In addition, the attention mechanism provides an effective way to make a recommender system more interpretable.

Although the above methods have achieved better performance, they still have some inherent limitations. First, review-level explanations can ensure the readability, but it is difficult for users to extract the accurate information from some tedious reviews. Second, due to users' arbitrary comments, the limited-quantity and low-quality aspect information is unsuitable for generating recommendation explanations. Finally, the most explainable recommendation methods usually exploit various attention networks to choose the explanation components, but the existing models [41,3,27] utilize the attention mechanism for a single recommendation explanation.

To this end, we propose a Triple-Attentional Explainable Recommendation model with Temporal Convolutional Network (TAERT) to capture the vector representation of user preferences and item features, and to learn the joint representation of user-item interactions based on different level attention networks for recommendation prediction and explanation. Specifically, the 1-D dilated causal convolution derives an exponentially large size of receptive field with the depth of the network, which enables TCN to learn both the neighbor feature interactions and global feature interactions. Furthermore, the fully convolutional network (FCN) architecture of TCN is specifically designed for learning the sequential characteristics. Hence, we propose to utilize TCN to learn useful features from the textual reviews. In addition to the capability of rating prediction, we also expect the proposed model to provide valuable explanations for users. Accordingly, we apply a review-level atten-

tion layer to select useful and representative reviews, which include detailed information and suggestions from other users. To help the users to understand the characteristics of the recommended items from the tedious reviews effectively, we also apply a word-level attention layer to identify informative words in the selected reviews. Due to the limited-quantity and low-quality aspect information in the textual reviews, we choose to apply a factor-level attention layer to exploit the abstract aspect information, and learn the interactions between user preferences and item features better. By adopting triple attention networks, we improve both the recommendation performance and the explainablity of the proposed model. The major contributions are summarized as follows:

- We propose a novel TCN feature learning method to learn word-aware and review-aware vector representations of user preferences and item features from the textual reviews, which fully considers the neighbor feature interactions, the global feature interactions and the sequential characteristics.
- The proposed TAERT model adopts the triple attention networks to enhance the quality of the learned features for better rating prediction. The explainability is also improved by providing review-level explanations with highlighted informative words.
- We demonstrate the higher rating prediction accuracy than the state-of-the-art methods by performing comprehensive experiments on six benchmark datasets. Moreover, by providing representative reviews with highlighted informative words, the case study shows the explainability of TAERT.

The rest of this paper is organized as follows: Related work is described in Section 2. Then, our framework is presented in Section 3. After that, the experimental results are shown in Section 4. Finally, brief conclusion and future work are given in Section 5.

2. Related work

In this section, we briefly discuss the related work in four different areas of rating prediction from textual reviews, attention based recommender systems, explainable recommender systems and the applications of temporal convolutional network.

2.1. Rating prediction from textual reviews

Although traditional MF methods are dominant strategies in recommender systems, two significant drawbacks were observed: data sparsity and the cold-start problem[20,30,21]. Therefore, a large amount of recent researches exploit the textual reviews to improve the performance of the rating prediction. Among them, CNN based text processor is widely adopted to learn the vector representation of users and items. For example, ConvMF [17] utilizes a CNN based network to learn the contextual features of review text and integrates the learned features into PMF for rating prediction. DeepCoNN [46] uses two parallel CNNs to learn the latent features of users and items from their textual reviews respectively, and then Factorization Machine (FM) is used for rating prediction. NRCA [25] proposes a CNN-based review encoder to extract the semantic features. TransNets [2] combines two CNN based networks, i.e., a source network and a target network to jointly predict the target user-item rating, which is proved to outperform DeepCoNN. CARP [22] proposes a CNN based extraction method and a sentiment capsule to identify the informative logic unit and the sentiment based representations for rating prediction.

2.2. Attention based recommender system

In recent years, the introduction of attention mechanism [40] significantly enhances the performance of recommendation. A large amount of works take different levels of attention networks into consideration, such as D-attn [36], A3NCF [6], NARRE [3], CARL [41], ACA-GRU [44] and RM-DRL [31]. Specifically, A3NCF proposes an aspect-aware attention network to capture users' specific aspect attentions on different items. CARL uses a word-level attention network to improve the performance of the model. NARRE extends DeepCoNN by an attention-based review pooling layer to model the review usefulness, and then presents a neural form LFM for predicting ratings. ACA-GRU leverages the attention mechanism in context-aware sequential recommendations, to distinguish the importance of various items in the rating sequence. RM-DRL proposes an attention-integrated gated recurrent unit to capture the different influences of different items in the preference history of the user final preferences. With attention mechanism, these methods not only perform much better than the traditional models but also facilitate the explainability of the recommendation.

2.3. Explainable recommender system

Actually, many methods have been developed to generate explanations based on the textual reviews in the last few years. Among them, TARMF [27] visualizes the user and item attention networks by highlighting the top-rated words of the textual reviews to enhance the transparency and explainability of their work. DER [4] and LUAR [15] provide review-level explanations that are highly rated by the review-level attention networks. EXPLORE [47] utilizes users' interests, item contents and

item tags to improve the rating prediction accuracy and provide tag-based explanations. LFA [49] exploits the user attributes and item attributes to generate a preference-based explanation automatically. DEAML [11] generates readable personalized explanations from its hierarchy with a dynamic programming algorithm. ELVis [10] utilizes the users' photos to predict attractive images of the recommended items for users and convince them of the qualities of the items. By providing various recommendation explanations, the scrutability, effectiveness, efficiency and persuasiveness of the recommendation, and the satisfaction of users may be improved [45].

2.4. TCN applications

Temporal Convolutional Network is designed for the sequential modeling tasks [1]. The FCN architecture ensures that the length of the input and output is kept the same. The dilated causal convolution enables TCN to dramatically increase the receptive field and fully consider the sequential information. Additionally, the residual module is also introduced to make the training process easier. Extensive experiments were conducted to indicate that TCN models substantially outperform other generic recurrent architectures such as LSTMs and GRUs. Recently, TCN based structure has been applied to many sequential modeling tasks, including scene text recognition [9], the speech enhancement [33], the stock trend prediction [8], El Niño-Southern Oscillation prediction [42] and recommender system [43]. Note that the difference of Hier-TCN [43] and our model TAERT is that we utilize TCN to learn the latent vector representations of users and items from the textual reviews, whereas the Hier-TCN uses TCN to model the temporal information of the user-item interactions.

3. The proposed model

In this section, we will introduce the details of our proposed model, Triple-Attentional Explainable Recommendation with Temporal Convolutional Network (TAERT). TAERT is devised to predict the rating for the given pair of user and item. The overall framework of TAERT is shown in Fig. 2. The model consists of two parallel neural networks: one for user modeling (NN_u) and another for item modeling (NN_i). On the top of the two networks, a factor-level attention based prediction layer is added to predict the rating for the given user and item. In the following subsections, we will first give a brief problem formulation. Then, we will present the proposed TCN feature learning method. After that, we will introduce the triple attention networks to adapt the TCN feature learning method to the rating regression task. Finally, we will detail the optimization process.

3.1. Problem formulation

For the training process of our model, we have a user set $\mathscr{U}$, an item set $\mathscr{I}$, the user-item rating matrix $\mathbf{R} \in \mathbb{R}^{|\mathscr{U}|\times|\mathscr{I}|}$ and the review text set $\mathscr{X}$. Given a user $u \in \mathscr{U}$ and an item $i \in \mathscr{I}$, the inputs of our model include the user review set $\mathscr{X}_u$, the item review set $\mathscr{X}_i$, the user ID embedding matrix $\mathbf{U}_i$ and the item ID embedding matrix $\mathbf{I}_u$.

For the outputs of our model, we provide a predicted rating $\hat{\mathbf{R}}_{u,i}$ and a review-level explanation with highlighted words. The notations with descriptions used in this paper are listed in Table 1. The two main tasks of our model can be summarized as: (1) learning the vector presentations of user preferences and item features from the textual reviews. (2) highlighting informative words in the textual reviews, selecting representative reviews and learning the different importance of latent factors for rating prediction and recommendation explanation.

3.2. TCN feature learning

In recent years, many CNN-based feature learning methods have been proposed and successfully applied into the recommender system, such as DeepCoNN [46], CARL [41] and NRPA [26], etc. However, when CNN is applied to learn the vector representations of the review, it ignores many useful global feature interactions and the word sequence of the textual reviews. To solve these problems, we propose a TCN feature learning method for better capturing user preferences and item features. Generally, the TCN feature learning method inputs a single text review and outputs an n-dimensional feature vector representation. Fig. 3(a) illustrates the component of the TCN feature learning.

3.2.1. Word embedding layer

The word embedding layer maps each word in the review into its l-dimensional vector representation, and transforms the given review into the review text matrix $\mathbf{X} \in \mathbb{R}^{l\times n}$ by concatenating the words in the preserved order and padding the review with zero to a fixed length n. The vector representations are expected to encode not only the lexical information but also semantic meaning from the text review. The word embedding layer is initialized with the pre-trained word vectors by word2vec [29], GloVe [34] or other embedding methods [16,14], and then fine-tuned by the back-propagation at the training stage. Note that the back-propagation on the embedding layer is beneficial to enriching the words in the dictionary, especially when the model is deployed online.

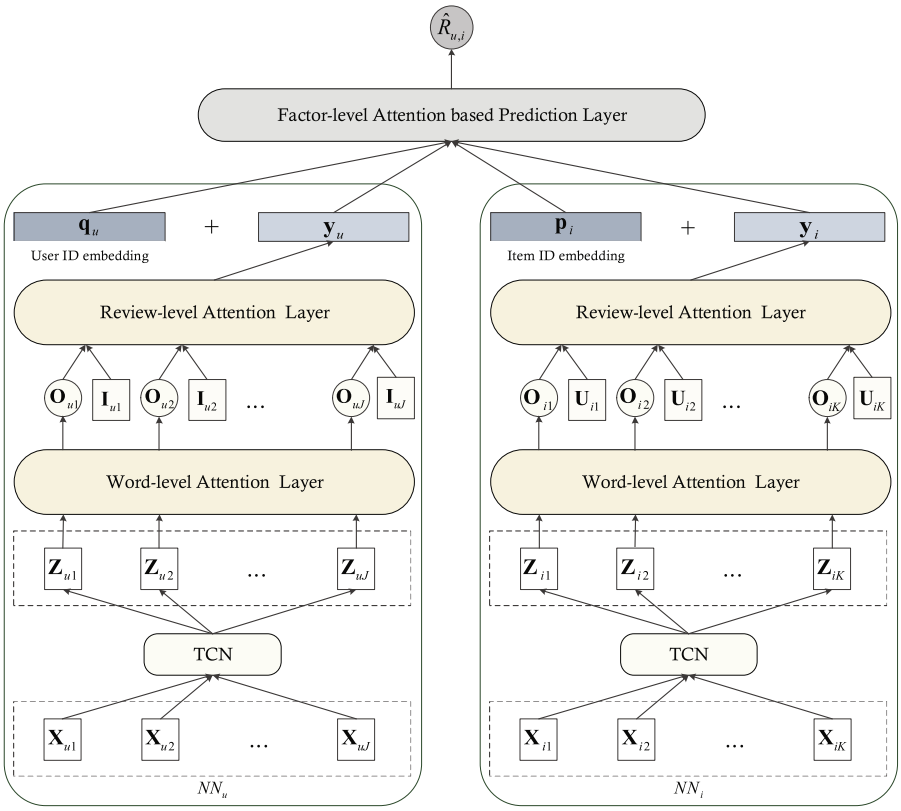

Fig. 2. The overall framework of TAERT.

Table 1
Notations.

Symbol	Description
$\mathscr{U}, \mathscr{I}$	The user set and the item set
$\mathbf{R}$	The user-item rating matrix
$\mathscr{X}$	The review text set
J	The fixed number of the reviews for each user
K	The fixed number of the reviews for each item
$\mathscr{X}_u$	The user review set that the user u ever wrote
$\mathscr{X}_i$	The item review set that was written for the item i
$\mathbf{U}_i$	The user ID embedding matrix who ever wrote a review for the item i
$\mathbf{I}_u$	The item ID embedding matrix that the user u ever wrote a review for
$\widehat{R}_{u,i}$	The predicted rating of user u for item i
$R_{u,i}$	The real rating of user u for item i
$\mathbf{X}$	The review text matrix
l	The dimension of the word embedding
n	The fixed length of the review text
$\mathbf{x}_j$	The jth word embedding vector in the review text $\mathbf{X}$
$\mathbf{z}_j$	The feature produced by the jth filter in the TCN layer
$\mathbf{z}_j^i$	The ith value in $\mathbf{z}_j$
$\mathbf{o}_j$	The jth value of the pooling layer
$\mathbf{W}_*, \mathbf{h}_*$	The weight matrix of each layer
$\mathbf{b}_*$	The bias of each layer
$\mathbf{a}_w^*$	The word-level attention scores
$\mathbf{a}_w$	The word-level attention weights
$\mathbf{a}_r$	The review-level attention weights
$\mathbf{h}_0$	The interaction of the user preference and item feature
$\mathbf{b}_u$	The user bias
$\mathbf{b}_i$	The item bias
μ	The global bias
$\mathbf{a}_f$	The factor-level attention weights

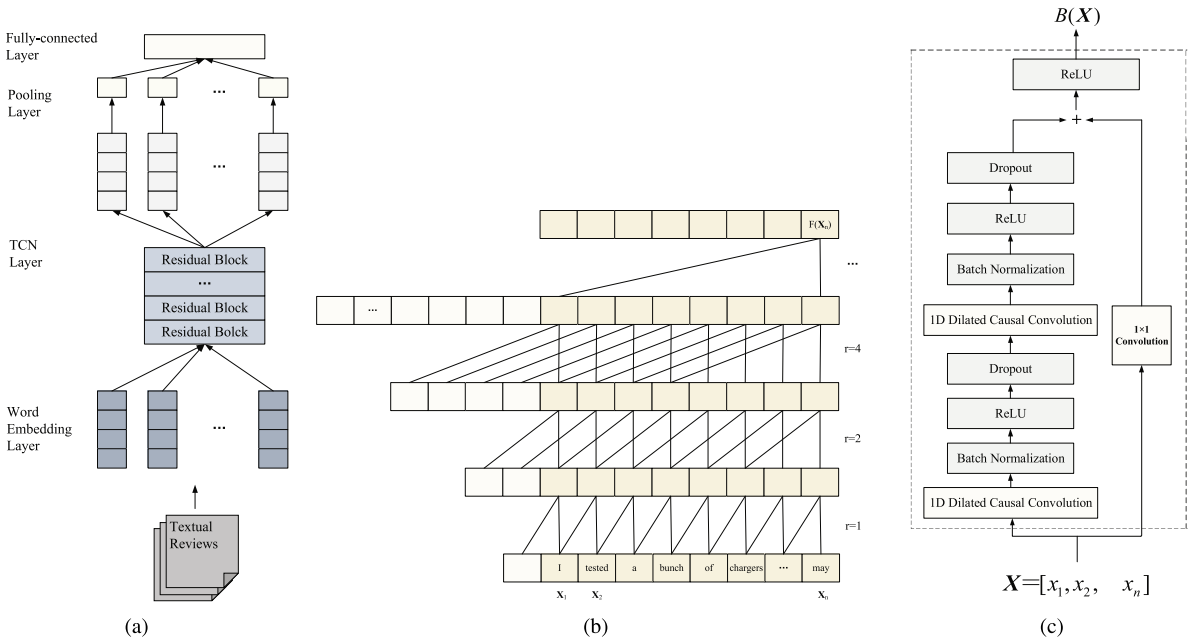

Fig. 3. Hierarchical description of review-based feature learning with TCN. (a) The overall framework of the proposed TCN feature learning method. (b) An example of 1D dilated causal convolution with kernel size $k = 2$. The output $F(\mathbf{X})$ keeps the same dimension as the input $\mathbf{X}$ for 'same' padding. The padded part is presented with a light-colored box. (c) The structure of the TCN residual block.

3.2.2. TCN layer

The TCN layer consists of m residual blocks. As is introduced in [1], each residual block is a two-layer network with 1-D dilated causal convolution, weight normalization, rectified linear unit (ReLU) and dropout. The structure of the residual block is shown in Fig. 3(c). As the core component of the TCN, 1-D dilated causal convolution can ensure to produce an output of the same length as input due to its FCN architecture. Furthermore, it can derive an exponentially large receptive field by increasing the dilation rate r with the depth of the network. Given a review text matrix $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ and t filters with each filter $f \in \mathbb{R}^{k \times l}$, the pth level 1-D dilated causal convolution operation F can be defined as:

$$r = 2^p, \quad 0 \leqslant p < m \; F(\mathbf{x}_j) = \sum_{i=0}^{k-1} f_i^T \mathbf{x}_{j-r \cdot i}, \quad \mathbf{x}_{\leq 0} := \mathbf{0} \; F(\mathbf{X}) = [F(\mathbf{x}_1), F(\mathbf{x}_2), \cdots, F(\mathbf{x}_n)] \tag{1}$$

where k denotes the kernel size and [] denotes the concatenation operation. Note that k^r zeros are padded at the top of the input $\mathbf{X}$ to keep $F(\mathbf{X})$ the same size as $\mathbf{X}$. An example of 1-D dilated causal convolution is shown in Fig. 3(b). It is obvious that increasing either kernel size k or dilation rate r can extend the receptive field. 1-D dilated causal convolution is beneficial to learn the sequential information of the words effectively. Moreover, when the dilation rate is big enough to make the receptive field cover all the words in the review, the global feature interaction can be easily captured.

As is shown in Fig. 3(c), the branch of the residual connections leads out to a series of transformations $S(\cdot)$. Let the 1×1 convolution operation be $C(\cdot)$ and then, the residual block operation $B(\cdot)$ on the review matrix $\mathbf{X}$ is defined as:

$$B(\mathbf{X}) = ReLU(C(\mathbf{X}) + S(\mathbf{X})) \tag{2}$$

For the input $\mathbf{X}$ of the TCN layer, the jth filter of the m residual blocks produces the features as:

$$\mathbf{z}_j = B_m(\cdots B_2(B_1(\mathbf{X}))\cdots) \tag{3}$$

The final output of the TCN layer is the concatenation of the outputs from its t filters, which is denoted by:

$$\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_t]^T \tag{4}$$

3.2.3. Pooling layer

For a single review, the jth filter of the TCN layer produces a word-aware feature $\mathbf{z}_j \in \mathbb{R}^n$. Then, a max-pooling operation is applied to capture the most important feature. The final output of the pooling layer is the concatenation of the outputs from its t filters. The pooling layer is formulated as:

$$\mathbf{o}_j = max\left(\mathbf{z}_j^1, \mathbf{z}_j^2, \cdots, \mathbf{z}_j^n\right) \ \mathbf{O} = [\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_t] \tag{5}$$

Generally, the most important word can't integrally express the meaning of the review. In TAERT, we introduce word-level attention layer to learn the different words' weight distribution instead of the max-pooling layer, which will be illustrated in Section 3.3.1. The method with max-pooling operation is taken as one of the baselines in the experiments.

3.2.4. Fully-connected layer

The final output of the TCN feature learning is generated by a fully-connected layer with a weight matrix $\mathbf{W} \in \mathbb{R}^{t \times n}$ and a bias $\mathbf{b} \in \mathbb{R}^n$, which is:

$$\mathbf{Q} = \mathbf{WO} + \mathbf{b} \tag{6}$$

where $\mathbf{Q}$ denotes the learned vector representation of the textual review X.

3.3. Triple attention networks

Attention mechanism has been widely introduced in many fields, such as recommender systems [24,26], speech recognition [35] and machine translation [5]. Inspired by the attention mechanism, triple attention networks are introduced into TAERT for rating prediction and recommendation explanation.

3.3.1. Word-level attention layer

As stated earlier, we assume that different words have different contributions to the feature vector representations. To this end, we introduce a word-level attention layer and define it as:

$$\mathbf{a}_w^* = \mathbf{h}_w^T tanh(\mathbf{W}_w\mathbf{Z} + \mathbf{b}_w) \tag{7}$$

where $\mathbf{W}_w \in \mathbb{R}^{n \times k_a}$ and $\mathbf{h}_w \in \mathbb{R}^{n \times k_a}$ are the weight matrices of the word-level attention layer, $\mathbf{b}_w \in \mathbb{R}^{k_a}$ is the bias of the word-level attention layer, k_a denotes the hidden layer size of the attention network and *tanh* is a nonlinear activation function.

Based on the above attention scores, we can highlight the word contribution in each review using the softmax function, which is defined as:

$$\mathbf{a}_w = \frac{exp(\mathbf{a}_w^*)}{\sum_{i=1}^{n} exp(\mathbf{a}_{wi}^*)} \tag{8}$$

Then, the final review-level feature vector is obtained by the following weighted sum:

$$\mathbf{O} = \sum_{i=1}^{n} \mathbf{a}_{wi}\mathbf{Z}_i \tag{9}$$

3.3.2. Review-level attention layer

To derive the feature vector representation of item i, we first process each review in the item review set $\mathcal{X}_i$ with the word-embedding layer, TCN layer and the word-level attention layer. Then, we aggregate the vector representations $\mathbf{O}_{i1}, \mathbf{O}_{i2}, \ldots, \mathbf{O}_{iK}$ to derive the item review feature $\mathbf{O}_i$. However, since low-quality reviews will further decrease the performance of the rating prediction and recommendation explanation, the review usefulness should be taken into consideration. Hence, we introduce attention mechanism to learn the weight distributions of all reviews. In addition to the item feature $\mathbf{O}_{i1}, \mathbf{O}_{i2}, \ldots, \mathbf{O}_{iK}$, the user ID embedding ($\mathbf{U}_i$) is also exploited to identify the useful reviews. Note that we also apply zero padding [3,24,25] to keep the number of reviews for each item with a fixed length K. To make attention weights easily comparable across different reviews, all weights are normalized for the attention weights by the softmax function:

$$\mathbf{a}_r = softmax\left(\mathbf{h}_r^T ReLU(\mathbf{W}_{r1}[\mathbf{O}_{i1}, \mathbf{O}_{i2}, \ldots, \mathbf{O}_{iK}] + \mathbf{W}_{r2}\mathbf{U}_i + \mathbf{b}_{r1}) + \mathbf{b}_{r2}\right) \tag{10}$$

where $\mathbf{W}_{r1} \in \mathbb{R}^{t \times k_a}, \mathbf{W}_{r2} \in \mathbb{R}^{l_i \times k_a}$ are weight matrices for the feature vector and the ID embedding, $\mathbf{h}_r \in \mathbb{R}^{k_a}$ denotes the global weight matrix, $\mathbf{b}_{r1} \in \mathbb{R}^{k_a \times K}, \mathbf{b}_{r2} \in \mathbb{R}^K$ represents local bias and global bias, l_i denotes the size of ID embedding, *ReLU* is a nonlinear activation function.

After that, we obtain the feature vector based on the attention weight of each review. Then, the final feature of item i is formed by a fully-connected layer as:

$$\mathbf{O}_i = \sum_{l=1}^{K} \mathbf{a}_{rl}\mathbf{O}_{il}\mathbf{y}_i = \mathbf{W}_0\mathbf{O}_i + \mathbf{b}_0 \tag{11}$$

where $\mathbf{W}_0 \in \mathbb{R}^{t \times n}, \mathbf{b}_0 \in \mathbb{R}^n$ denote the weight matrix and bias of the fully-connected layer, respectively.

3.3.3. Factor-level attention based prediction layer

To get the predicted rating $\widehat{R}_{u,i}$, we use the neural form LFM [13] to capture the high-order nonlinear feature interactions as:

$$\widehat{R}_{u,i} = \mathbf{W}_1^T \mathbf{h}_0 + \mathbf{b}_u + \mathbf{b}_i + \mu \tag{12}$$

where $\mathbf{W}_1 \in \mathbb{R}^n$ denotes the weight matrix of the prediction layer, $\mathbf{h}_0$ denote the interaction of the user preference and item feature, $\mathbf{b}_u, \mathbf{b}_i$ and μ denotes the user bias, item bias and global bias, respectively.

To model the interaction of user preference and item feature, we first extend them to two components: one based on rating information and the other based on textual reviews. Then, the latent factors of users and items are mapped into a shared hidden space. Finally, we introduce the weight of latent factors obtained from the attention network to model the importance of different latent factors. The overall process can be formulated as:

$$\mathbf{h}_0 = \mathbf{a}_f \odot (\mathbf{q}_u + \mathbf{y}_u) \odot (\mathbf{p}_i + \mathbf{y}_i) \tag{13}$$

where $\mathbf{a}_f$ denotes the learned attention weight of different latent factors, $\mathbf{q}_u$ and $\mathbf{p}_i$ are the vector representations of the user preferences and item features based on ratings, $\mathbf{y}_u$ and $\mathbf{y}_i$ are the vector representations of the user preferences and item features learned from the method introduced in Section 3.3.2, and $\odot$ denotes the element-wise product.

Inspired by [6], we model the importance of different factors, i.e., abstract aspect information, by using the concatenation of $\mathbf{q}_u, \mathbf{p}_i, \mathbf{y}_u$ and $\mathbf{y}_i$ as input:

$$\mathbf{a}_f = softmax\Big(\mathbf{h}_f^T ReLU(\mathbf{W}_f[\mathbf{q}_u, \mathbf{p}_i, \mathbf{y}_u, \mathbf{y}_i] + \mathbf{b}_f)\Big) \tag{14}$$

where $\mathbf{W}_f \in \mathbb{R}^{4n \times k_a}$ and $\mathbf{h}_f \in \mathbb{R}^{n \times k_a}$ represent the local weight matrix and the global weight matrix, respectively, and $\mathbf{b}_f \in \mathbb{R}^{k_a}$ denotes the bias of the attention layer.

3.4. Model optimization

As the rating prediction is a regression task in essence, the learning of TAERT is the same as [6,11,41] with square loss. The objective function is defined as:

$$\mathscr{L}_{sqr} = \sum_{(u,i)\in\mathscr{D}} \left(\widehat{R}_{u,i} - R_{u,i}\right)^2 + \lambda_\Theta \|\Theta\|^2 \tag{15}$$

where $\mathscr{D}$ denotes the set of observed user-item rating pairs in the training set, $R_{u,i}$ is the real rating of user u for item i, $\widehat{R}_{u,i}$ is the prediction rating, $\|\Theta\|^2$ denotes the L2 norm of all the parameters in our model, and λ_Θ denotes the weight of $\|\Theta\|^2$. Adaptive Moment Estimation (Adam) [18] is adopted for optimization, whose main advantage is automatic learning rate tuning, fast training process and stable gradient descent. Also, the dropout method [37] is applied to alleviate overfitting at the training stage.

4. Experiments

In this section, We conduct comprehensive experiments to evaluate the performance of our proposed model in both rating prediction and recommendation explanation.

4.1. Experimental settings

4.1.1. Datasets

In our experiments, we exploit six publicly available datasets from different domains. The six datasets are collected from Amazon 5-core[1] [12]: Office Products, Digital Music, Baby, Toys and Games, Cell Phones and Accessories, Clothing, Shoes and Jewelry. Both users and items have at least 5 reviews, and users' ratings on item are ranging from 1 to 5. To alleviate the long tail effect of reviews, we follow the preprocessing steps in [3] to adjust the length and the number of the reviews. The statistics of the six datasets are shown in Table 2.

For each dataset, we randomly split it into training set (80%), validation set (10%) and testing set (10%). The validation set is used to tune the hyper-parameters. The performance of comparison results derives from the testing set. Note that we only exploit the review information at the training stage, because reviews are unavailable at the validation and testing stage in the real world.

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

Table 2
Statistics of the six datasets.

Dataset	#Users	#Items	#Ratings & Reviews	Density
Office Products	4,905	2,420	53,258	0.448%
Digital Music	5,541	3,568	64,706	0.327%
Baby	19,445	7,050	160,792	0.117%
Toys & Games	19,412	11,924	167,597	0.072%
Cell Phones & Accessories	27,879	10,429	194,439	0.066%
Clothing, Shoes & Jewelry	39,387	23,033	278,677	0.030%

4.1.2. Baselines

For the purpose of comparison, we evaluate our proposed model TAERT with eight state-of-the-art models. The characteristics of the comparative models are given in Table 3.

- **PMF** [30]: Probabilistic Matrix Factorization. It introduces Gaussian distribution to model the latent factors for users and items.
- **NeuMF** [13]: Neural network based Collaborative Filtering. It uses neural networks to model the interactions between the latent factors for users and items to predict the rating.
- **ConvMF** [17]: Convolutional Matrix Factorization. It employs CNN to learn the contextual features of the reviews, and then integrates them into PMF for rating prediction.
- **DeepCoNN** [46]: Deep Cooperative Neural Network. By jointly modeling users and items with two parallel CNN networks, it extracts the latent feature vectors from the textual reviews, and FM is then used for rating prediction.
- **MPCN** [38]: Multi-Pointer Co-attention Networks for recommendation. It proposes a novel pointer-based co-attention mechanism to learn the review usefulness and the word-level interaction. Also, a multi-pointer learning scheme and FM are utilized to predict ratings.
- **CARL** [41]: Context-Aware user-item Representation Learning. It exploits CNN and attention mechanism to derive pair-dependent latent representations, and introduces a dynamic linear fusion for rating prediction.
- **NARRE** [3]: Neural Attentional Rating Regression with Explanations. It utilizes two parallel CNNs and attention mechanism to learn the latent features and model the review usefulness. A neural form LFM is presented for predicting ratings.
- **DAML** [24]: Dual Attention Mutual Learning. It utilizes local and mutual attention of the CNN to jointly learn the features, and a neural FM is introduced to predict ratings.

4.1.3. Evaluation metric

To evaluate the performance of all the models, we adopt the well-known Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) as evaluation metrics:

$$RMSE = \sqrt{\frac{1}{|\mathscr{D}_t|}\sum_{(u,i)\in\mathscr{D}_t}\left(\widehat{R}_{u,i} - R_{u,i}\right)^2} \quad MAE = \frac{1}{|\mathscr{D}_t|}\sum_{(u,i)\in\mathscr{D}_t}|\widehat{R}_{u,i} - R_{u,i}| \tag{16}$$

where $\mathscr{D}_t$ is the set of user-item pairs in the testing set. Note that the lower RMSE and MAE indicate better performance.

4.1.4. Hyper-parameter settings

For fair comparison of the performance, we initialize the parameters of all the baselines based on the setting strategies reported in their papers, and then we tune the parameters carefully to achieve the optimal performance. For DeepCoNN and CARL, we apply Adam optimizer instead of the RMSprop [39] optimizer to get better performance. For deep learning based models, the learning rate is tuned from $[0.001, 0.002, 0.005, 0.01]$. The dropout ratio is searched in $[0.1, 0.2, 0.3, 0.4, 0.5]$. The number of the latent factor is optimized from $[8, 16, 32, 64, 128]$ and the number of the filters is tested in $[50, 100, 150, 200, 250]$. The batch size is set to 10.

For TAERT, a pre-trained word embedding method [29] is used in this paper. The length of the convolution window is $k = 3$, and the number of the filters t is set to 100. The attention size k_a is set to 32. The dropout ratio is set to 0.1 and the number of latent factors is set to 32. Moreover, the regularization parameter is set to 0.01.

4.2. Overall performance analysis

For all the models, we train them till convergence and then report the best test result. The final comparison results of our model TAERT with the baselines on the Amazon 5-core datasets are shown in Table 4 and Table 5, which take RMSE and MAE as evaluation metrics, respectively. After analyzing the results, we make the following conclusions:

First, PMF shows the worst performance on six datasets, especially on the sparser datasets like Toys & Games, Cellphones & Accessories, Clothing, Shoes & Jewelry. It indicates that deep learning methods can model high-level user preferences and item features more effectively than traditional methods with non-linear interactions.

Table 3
Comparison of the Models.

Characteristics	PMF	NeuMF	ConvMF	DeepCoNN	MPCN	CARL	NARRE	DAML	TAERT
Ratings	√	√	√	√	√	√	√	√	√
Reviews	/	/	√	√	√	√	√	√	√
Textual Feature Learning Method	/	/	CNN	CNN	gating	CNN	CNN	CNN	TCN
Word Contribution	/	/	/	/	√	√	/	√	√
Review Usefulness	/	/	/	/	√	√	√	/	√
Factor Importance	/	/	/	/	/	/	/	/	√

Table 4
Overall Performance comparison on six datasets for all methods in terms of RMSE. The best and second best results are highlighted in boldface and underlined, respectively. Δ% denotes the performance improvement of TAERT over the best baseline.

RMSE	Office Products	Digital Music	Baby	Toys & Games	Cellphone & Accessories	Clothing, Shoes & Jewelry
PMF	1.0673	1.0634	1.3935	1.3992	1.6084	1.6137
NeuMF	0.9062	0.9993	1.2525	1.0424	1.3232	1.2144
ConvMF	0.8913	0.9395	1.1369	0.9311	1.2272	1.1058
DeepCoNN	0.8573	0.8943	1.0807	0.8984	1.1972	1.0590
MPCN	0.8790	0.9746	1.1189	0.9588	1.1857	1.0886
CARL	0.8564	0.9479	1.0972	0.9368	1.2069	1.0884
DAML	0.8558	0.9128	1.1035	0.9287	1.1669	1.0793
NARRE	**0.8437**	0.8886	1.0711	0.8881	**1.1527**	1.0475
TAERT	0.8447	**0.8847**	**1.0710**	**0.8824**	1.1549	**1.0459**
Δ%	-0.11	0.43	0.01	0.64	-0.19	0.15

Table 5
Overall Performance comparison on six datasets for all methods in terms of MAE. The best and second best results are highlighted in boldface and underlined, respectively. Δ% denotes the performance improvement of TAERT over the best baseline.

MAE	Office Products	Digital Music	Baby	Toys & Games	Cellphone & Accessories	Clothing, Shoes & Jewelry
PMF	0.8116	0.8049	1.1113	1.1062	1.2939	1.3172
NeuMF	0.7111	0.7664	0.9911	0.8044	1.0515	0.9723
ConvMF	0.6356	0.6725	0.8431	0.6760	0.9251	0.8292
DeepCoNN	0.6246	0.6617	0.8139	0.6443	0.8957	0.7889
MPCN	0.6644	0.7351	0.8507	0.6964	0.9183	0.8326
CARL	0.6373	0.7426	0.8569	0.7257	0.8979	0.8922
DAML	0.6141	0.6642	0.8287	0.6660	0.9123	0.8055
NARRE	0.6181	0.6530	0.8034	0.6393	0.8776	0.7865
TAERT	**0.6088**	**0.6358**	**0.7806**	**0.6173**	**0.8601**	**0.7606**
Δ%	0.86	2.63	2.83	3.44	1.99	3.29

Second, the methods with exploiting review information generally perform much better than those only relying on a rating matrix. It should be attributed to the CNN-based or TCN-based architecture for textual feature learning, because the semantic information can be easily captured for the high-quality vector representations of user preferences and item features. Among the review-based models, MPCN usually delivers a bad performance. One possible explanation is that the gating mechanism cannot capture useful features effectively.

Third, the methods that utilize attention mechanism generally perform better than others, which demonstrates that the attention mechanism does help CNN to capture relevant information of user preferences and item features from the ample reviews. However, though CARL introduces a word-level attention network, compared with DeepCoNN model, it usually obtains worse performance. One possible reason is that the network of CARL is not deep enough.

Finally, our TAERT model achieves the best RMSE score on four datasets and the best MAE scores on six datasets. Moreover, we can observe that the average RMSE and MAE improvement over the best baseline are 0.16% and 2.51%, respectively. It suggests that the proposed model can achieve comparable performance in terms of RMSE and outperform the state-of-the-art baselines like NARRE and DAML in terms of MAE. In other words, to enable recommendation explainability, we do not sacrifice but even improve the recommendation performance. What's more, it verifies that the proposed TCN feature learning method and the triple attention networks are effective for rating prediction.

4.3. Stability analysis

In this subsection, we will analyze the stability of NARRE and the proposed model TAERT, by spliting the dataset with different ratio of the training set to testing set. The results of the experiments are shown in Fig. 4. Due to the space limitation,

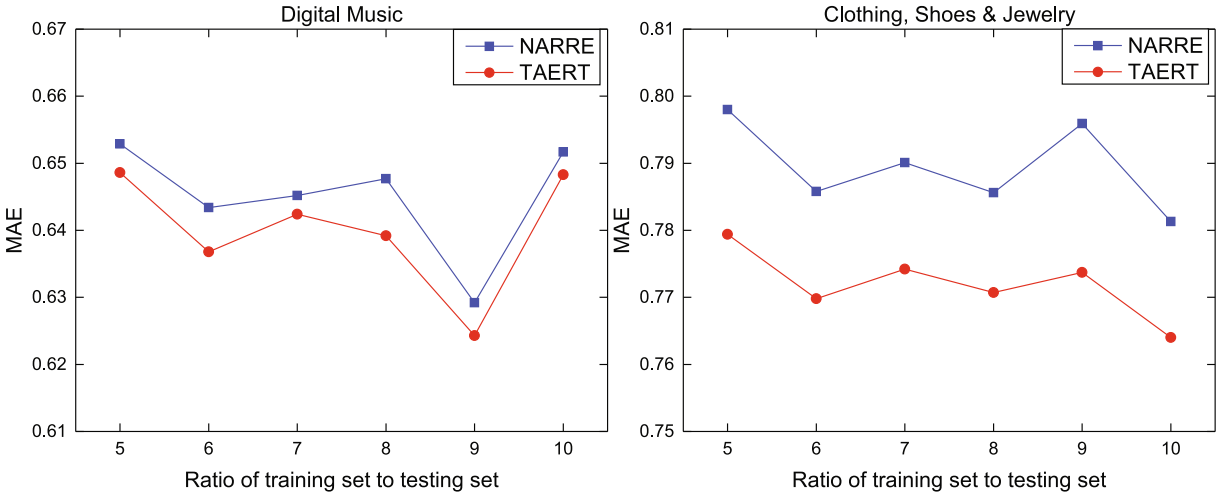

Fig. 4. Stability analysis with varying ratio of training set to testing set.

we choose a small dataset, Digital Music, and a big dataset, Clothing, Shoes & Jewelry for presentation. From the Fig. 4, we can find that the results of Digital Music are more sensitive to the ratio of the training set to testing set than the results of the Clothing, Shoes & Jewelry. One possible explanation is that the first dataset is very small, which makes the performance more likely to depend on the dataset split. Furthermore, we can find that TAERT outperforms NARRE on both datasets in the range of [5–10], which also verifies the superiority of TAERT.

4.4. Parameter sensitivity study

In this subsection, we mainly analyze the sensitivity about the number of the latent factors and the number of filters on DeepCoNN, NARRE and the proposed method TAERT. The results of the experiments are shown in Figs. 5 and 6.

First, we discuss the impact about the number of the latent factors. As seen from Fig. 5, with the change of the number of the latent factors, the performance of the TAERT model varies very little. Moreover, TAERT outperforms DeepCoNN and NARRE on both datasets in the range of $[8, 16, 32, 64, 128]$, which demonstrates that TAERT isn't sensitive to the number of the latent factors.

In terms of the number of filters, TAERT also performs much better than DeepCoNN and NARRE on both datasets in the range of $[50, 100, 150, 200, 250]$. Furthermore, we can see that TAERT isn't sensitive to the number of filters, which is attributed to the proposed TCN feature learning method.

4.5. Effectiveness analysis of TCN feature learning method

In this subsection, we will further explore the effectiveness of the proposed TCN feature learning method. To better demonstrate its superiority, we compare it with CNN, LSTM, C-LSTM [48] and Transformer [40] based textual feature learning methods. Note that the word-level attention network of TAERT requires the learned features to be word-aware, but the CNN, LSTM, C-LSTM based textual feature learning methods cannot ensure that. Hence, we take the NARRE as the baseline to demonstrate the effectiveness of the TCN feature learning method. We first replace the CNN of the NARRE with LSTM and C-LSTM, and name them NARRE-LSTM, NARRE-C-LSTM, respectively. Then, we extend NARRE by replacing its CNN text processor with TCN feature learning method and Transformer network, and name it NARRE-TCN, NARRE-Transformer, respectively. Note that the Transformer architechture is originally designed for sequence-to-sequence learning tasks with an encoder-decoder structure. Hence, we only adopt the encoder of the Transformer, which consists mainly of positional encoding, multi-head self-attention networks and position-wise fully connected feed forward networks. The results of them on six datasets are shown in Table 6.

From the Table 6, it can be observed that NARRE-TCN achieves the best MAE scores on five datasets. Among the baselines, CNN is capable of learning neighbor features but ignores the global features and the sequential correlations. LSTM is designed for equential modeling, but it cannot learn features in a parallel way. The C-LSTM integrates their advantages by replacing the pooling operation in the CNN with LSTM. The Transformer network is solely based on self-attention mechanism and dispenses with convolutions and recurrence entirely, which is widely applied in natural language processing tasks. However, it also makes the model unable to capture useful neighbor features. Furthermore, although it introduces the positional embedding to inject the absolute position information, it does not inherently solve the problem of sequential information loss. In contrast, the proposed TCN feature learning method can reap the benefits by capturing the neighbour feature interactions, the global feature interactions and the word sequence characteristics from the textual reviews. Also,

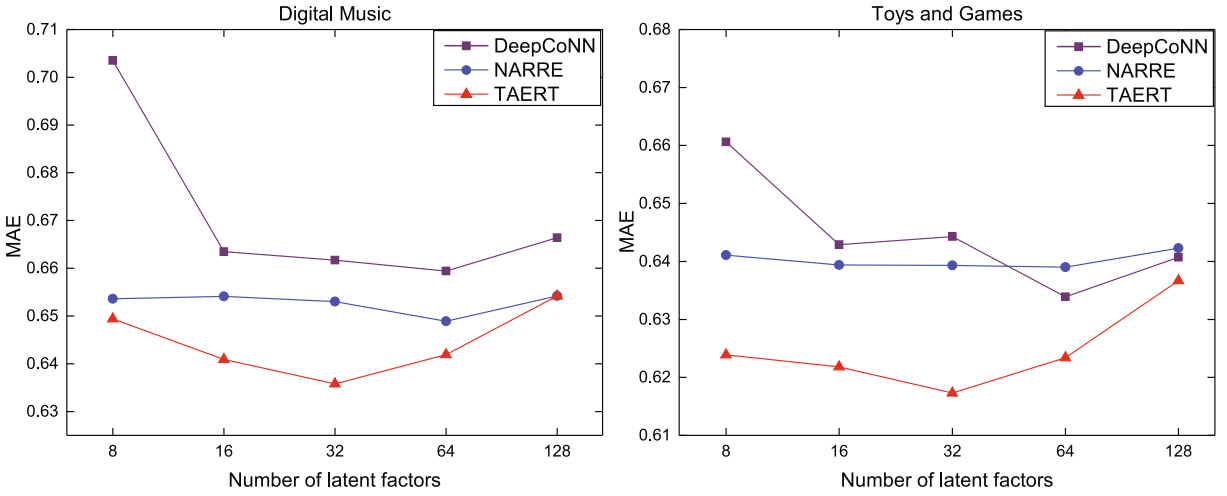

Fig. 5. Parameter sensitivity study on the number of latent factors.

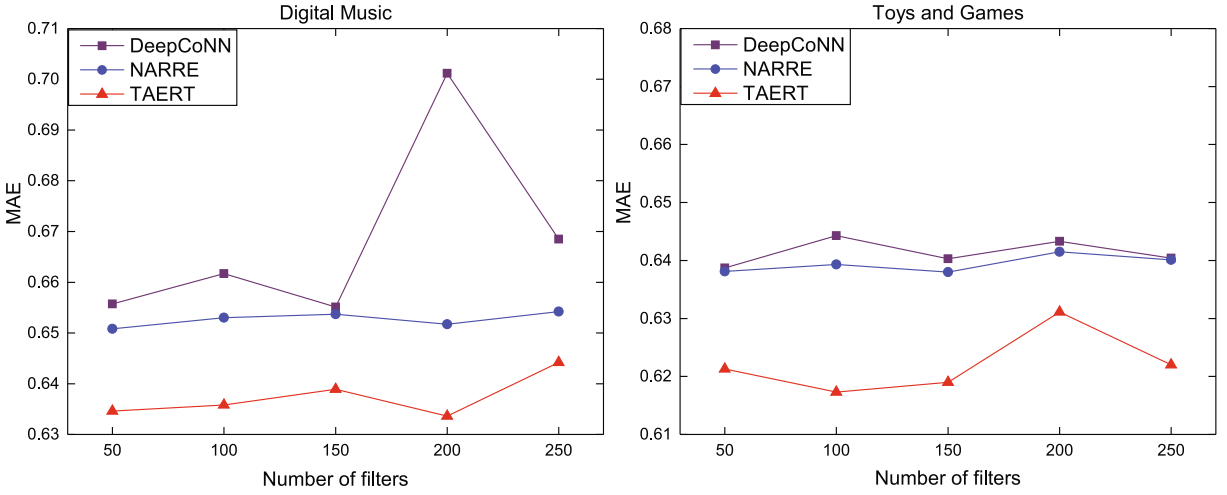

Fig. 6. Parameter sensitivity study on the number of filters.

Table 6
The effect of TCN feature learning method in terms of MAE on six datasets.

MAE	Office Products	Digital Music	Baby	Toys & Games	Cellphone & Accessories	Clothing, Shoes & Jewelry
NARRE	0.6181	0.6523	0.8047	0.6419	0.8776	0.7865
NARRE-LSTM	0.6218	0.6557	0.8108	0.6362	0.8855	0.7852
NARRE-C-LSTM	0.6189	0.6540	0.8038	0.6389	0.8793	0.7842
NARRE-Transformer	0.6153	0.6533	0.8027	0.6339	0.8749	0.7857
NARRE-TCN	0.6183	0.6457	0.8005	0.6344	0.8734	0.7840
TAERT	0.6088	0.6314	0.7966	0.6193	0.8601	0.7606

the FCN architecture of TCN ensures the learned features to be word-aware, which further enables us to introduce word-level attention network. Therefore, the user preferences and item features can be well learned for higher prediction accuracy. What's more, we can observe that TAERT derives the best performance against the others. It demonstrates the benefits of the triple attention networks.

4.6. Effectiveness analysis of triple attention networks

In this subsection, we further analyze the effectiveness of triple attention networks: word-level attention layer, review-level attention layer and factor-level attention based prediction layer.

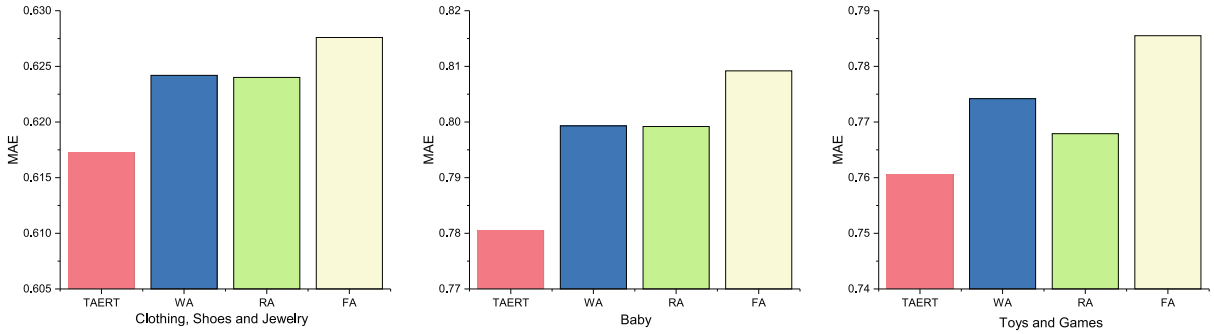

Fig. 7. The impact of attention network of each level in terms of MAE on three datasets.

Recall that, we introduce a max-pooling operation and analyze its defectives. Here we adopt it as a baseline and name it WA. Similarly, we utilize the mean-pooling strategy instead of review-level attention layer in TAERT and name it RA. To analyze the impact of the factor-level attention based prediction layer, we simply remove it and assign a normalized constant weight to each latent factor, and name it FA. The final results of the experiments are shown in Fig. 7.

From the Fig. 7, we can see that TAERT performs much better than the other baselines on the three datasets. This demonstrates that attention mechanism improves the performance of rating prediction. What's more, FA achieves the worst performance among the three baselines. A primary reason is that when modeling the users and items, the consideration of different importance of latent factors is more significant. It also indicates that the abstract aspect information is conducive to improving the performance of TAERT.

4.7. Explainabality case study

In this subsection, we discuss the explainability of TAERT. As stated in the previous works [3,4], a large amount of information and suggestions covered in the textual review are beneficial for users to make faster and better decisions. Accordingly, we also provide review-level explanations to improve the explainability of TAERT. Additionally, TAERT is capable of highlighting the informative words in the provided review to help users understand the item features better.

Fig. 8 shows a case study of how TAERT recommends items to users, generates review-level explanations and highlights the informative words. The experiments are conducted on the dataset of Cellphone & Accessories. According to the predicted rating, we recommend the top-3 rated items to the user #A10DHJK4D0QFKR. On the basis of the review-level attention weights, we select the top-3 rated user reviews to demonstrate the user preference, and generate the corresponding review-level explanation with the top-1 rated item review. From the figure, we can see that the reviews selected by TAERT are highly instructive and informative for users to get the characteristics of the recommended item. Moreover, TAERT performs well in learning the user preferences, item features and their interactions from the textual reviews.

For example, the user reviewed that, "I use it as my main charger because of the USB plug". Based on this, TAERT recommends the top-1 grad item #B004OZMWUS because the item review holds that, "I tested a bunch of chargers with my Samsung Galaxy S3 and the best chargers were the following in this order. . .However, there are other advantages to the Blackberry Playbook charger which are a six foot cord and the USB cord cannot be detached from the plug. . .". This means that the recommended charger is equipped with USB cord that can't be detached from the plug. It indicates that the user preference in USB plug is well learned and applied for the recommendation. Besides, significant similarity between the user preferences and item features is also presented by the other two examples.

To help users understand the item features better, we highlight the highly-rated words in the review explanation by the word-level attention layer. The heat map of the review is visualized in Fig. 8. From the figure, we can see that the words such as "charger", "phone", "USB" and "plug" are highlighted in the first item review, which include the basic characteristics of the recommended item. Similar results are also shown by the other two examples. To sum up, the explainability of TAERT is significantly improved by introducing word-level attention layer and review-level attention layer. The former can highlight informative words in the textual review while the latter can select representative reviews. Also, this case study verifies the superiority of TAERT's explainability.

5. Conclusion and future work

In this paper, we propose a triple-attentional explainable recommendation model with temporal convolutional network. By applying TCN feature learning, the vector representations of user preferences and item features are well learned from the given textual reviews. Besides, word-level attention layer, review-level attention layer and the factor-level attention based prediction layer are unified to model the interactions between user preferences and item features for better rating predic-

	Item_ID	Review-level explanation
Top-1	#B004OZMWUS	I tested a bunch of chargers with my Samsung Galaxy S3 and the best chargers were the following in this order:#1) HP Touchpad charger; #2) Blackberry Playbook charger; #3) Asus Nexus 7 charger; #4) Samsung cube charger 1 amp; #5) Samsung oblong charger .7 amps; #6) New Trent wall charger with premium Amazonbasics cord- This one was the absolute worst and only good for overnight charging. I only measured about a 10% improvement in charger time over the Oblong charger that came with the phone. However, there are other advantages to the Blackberry Playbook charger which are a six foot cord and the usb cord cannot be detached from the plug...
Top-2	#B003CK70VC	My purpose for buying this cable was to hook my phone up to the axillary input in my car so I could stream Pandora though my car's stereo. Before I bought this cable I tried and old stereo cable (use for a camera trigger) in my closet. Well that cable worked, but the end of the connector (just before the 1/8\" plug) was squared off and not recessed. This prevented the cable from being inserted fully into my phone (Samsung Infuse).This is actually a common problem for many older stereo cables. Their larger diameter prevents them from being fully compatible with newer devices which have a smaller diameter countersink just before the plug...
Top-3	#B002YFDRHW	I ordered this external battery pack/charger to charge multiple items, and it works with every one of them. I have a Verizon MiFi 2200 wireless antenna, which does NOT have the greatest battery life. This was the #1 item I wanted the Trent IMP880 for, and it works like a charm! The IMP 880 comes with many different adaptor \"plugs\" to fit various items. The IMP880 charges my iPad (and it also included a dedicated iPad charging cable), my Nokia cellphone, my iPod Nano 4th, and the Verizon MiFi. I used the IMP880 recently on a trip to recharge my iPad when it was at only 10% power, and then had plenty of power to recharge my iPod, my cell phone and my Verizon MiFi which were all at about 1/4 battery life (estimating on the MiFi, 'cause you can't tell!)...

Fig. 8. Explainability Case Study in Cellphones & Accessories. Top-3 recommended item for user#A10DHJK4D0QFKR is given, together with the review-level explanation. The review segment at the bottom is extracted from the top-3 informative user reviews to represent the user preferences.

tion. We conduct a series of experiments to verify both the performance of the rating prediction and the explainability of TAERT. Specifically, we compare the TCN feature learning method with CNN, LSTM, C-LSTM and Transformer based feature learning methods to prove the superiority of TCN in modeling textual reviews. We also remove the factor-level attention layer to demonstrate how the recommendation performance deteriorates without abstract aspect information. A case study of recommendation explanation is given to show how TAERT helps users to fully and quickly understand the item characteristics from the tedious reviews.

However, only providing review-level explanations is far from what the human beings expect. What we really need is to enable the machine to write informative and readable explanations by itself. Recent works like [32,19] utilize deep neural networks and knowledge graph to generate natural language from the given context. As a future work, we will explore the possibility to generate summary-level explanation from the item reviews for better explainability.

CRediT authorship contribution statement

Siyuan Guo: Methodology, Software, Writing - original draft. **Ying Wang:** Conceptualization, Resources, Project administration. **Hao Yuan:** Software, Investigation. **Zeyu Huang:** Validation. **Jianwei Chen:** Data curation. **Xin Wang:** Writing - review & editing, Supervision, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We truly thank the reviewers for pertinent comments. This work was supported by a grant from the National Natural Science Foundation of China under grants (Nos. 61872161 and 61976103), and the China Postdoctoral Science Foundation (No. 2017M611301), and the Nature Science Foundation of Jilin Province (Nos. 20200201297JC and 2018101328JC), and

the Foundation of Development and Reform of Jilin Province (No. 2019C053-8), and the Foundation of Jilin Educational Committee (No. JJKH20191257KJ) and the Fundamental Research Funds for the Central Universities, JLU.

References

[1] Shaojie Bai, J. Zico Kolter, Vladlen Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, arXiv preprint arXiv:1803.01271, 2018..
[2] Rose Catherine, William Cohen, Transnets: Learning to transform for recommendation, in: Proceedings of the Eleventh ACM Conference on Recommender Systems, ACM, 2017, pp. 288–296.
[3] Chong Chen, Min Zhang, Yiqun Liu, Shaoping Ma, Neural attentional rating regression with review-level explanations, in: Proceedings of the 2018 World Wide Web Conference, International World Wide Web Conferences Steering Committee, 2018, pp. 1583–1592..
[4] Xu. Chen, Yongfeng Zhang, Zheng Qin, Dynamic explainable recommendation based on neural attentive models, Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 53–60.
[5] Yong Cheng, Agreement-based joint training for bidirectional attention-based neural machine translation, in: Joint Training for Neural Machine Translation, Springer, 2019, pp. 11–23.
[6] Zhiyong Cheng, Ying Ding, Xiangnan He, Lei Zhu, Xuemeng Song, Mohan S. Kankanhalli, A 3ncf: An adaptive aspect attention model for rating prediction, in: IJCAI, 2018, pp. 3748–3754..
[7] Zhiyong Cheng, Ying Ding, Lei Zhu, Mohan Kankanhalli, Aspect-aware latent factor model: Rating prediction with ratings and reviews, in: Proceedings of the 2018 World Wide Web Conference, International World Wide Web Conferences Steering Committee, 2018, pp. 639–648..
[8] Shumin Deng, Ningyu Zhang, Wen Zhang, Jiaoyan Chen, Jeff Z. Pan, Huajun Chen, Knowledge-driven stock trend prediction and explanation via temporal convolutional network, in: Companion Proceedings of The 2019 World Wide Web Conference, WWW '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 678–685.
[9] X. Du, T. Ma, Y. Zheng, H. Ye, X. Wu, L. He, Scene text recognition with temporal convolutional encoder, in: ICASSP 2020 – 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 2383–2387.
[10] Jorge Díez, Pablo Pérez-Núñez, Oscar Luaces, Beatriz Remeseiro, Antonio Bahamonde, Towards explainable personalized recommendations by learning from users' photos, Information Sciences 520 (2020) 416–430.
[11] Jingyue Gao, Xiting Wang, Yasha Wang, Xing Xie, Explainable recommendation through attentive multi-view learning, in: AAAI, 2019..
[12] Ruining He, Julian McAuley, Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering, in: Proceedings of the 25th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2016, pp. 507–517..
[13] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, Tat-Seng Chua, Neural collaborative filtering. In Proceedings of the 26th international conference on world wide web, International World Wide Web Conferences Steering Committee, 2017, pp. 173–182..
[14] Eric H. Huang, Richard Socher, Christopher D. Manning, Andrew Y. Ng, Improving word representations via global context and multiple word prototypes, in: Meeting of the Association for Computational Linguistics: Long Papers, 2012..
[15] Dongmin Hyun, Chanyoung Park, Junsu Cho, Yu Hwanjo, Learning to utilize auxiliary reviews for recommendation, Information Sciences 545 (2021) 595–607.
[16] Armand Joulin, Edouard Grave, Piotr Bojanowski, Tomas Mikolov, Bag of tricks for efficient text classification, arXiv preprint arXiv:1607.01759, 2016..
[17] Donghyun Kim, Chanyoung Park, Oh Jinoh, Sungyoung Lee, Hwanjo Yu, Convolutional matrix factorization for document context-aware recommendation, in: Proceedings of the 10th ACM Conference on Recommender Systems, ACM, 2016, pp. 233–240.
[18] Diederik P. Kingma, Jimmy Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980, 2014..
[19] Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, Hannaneh Hajishirzi, Text generation from knowledge graphs with graph transformers, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 2284–2293.
[20] Yehuda Koren, Robert Bell, Chris Volinsky, Matrix factorization techniques for recommender systems, Computer 8 (2009) 30–37.
[21] Daniel D. Lee, H. Sebastian Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (6755) (1999) 788.
[22] Chenliang Li, Cong Quan, Li Peng, Yunwei Qi, Yuming Deng, Wu. Libing, A capsule network for recommendation and explaining what you like and dislike, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2019, pp. 275–284.
[23] Greg Linden, Brent Smith, Jeremy York, Amazon. com recommendations: Item-to-item collaborative filtering, IEEE Internet Computing (1) (2003) 76–80..
[24] Donghua Liu, Jing Li, Du. Bo, Jun Chang, Rong Gao, Daml: Dual attention mutual learning between ratings and reviews for item recommendation, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, 2019, pp. 344–352.
[25] Hongtao Liu, Wenjun Wang, Xu. Hongyan, Qiyao Peng, Pengfei Jiao, Neural unified review recommendation with cross attention, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Association for Computing Machinery, New York, NY, USA, 2020.
[26] Hongtao Liu, Fangzhao Wu, Wenjun Wang, Xianchen Wang, Pengfei Jiao, Chuhan Wu, Xing Xie, Nrpa: Neural recommendation with personalized attention, in: Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'19, ACM, New York, NY, USA, 2019, pp. 1233–1236..
[27] Yichao Lu, Ruihai Dong, Barry Smyth, Coevolutionary recommendation model: Mutual learning between ratings and reviews, in: Proceedings of the 2018 World Wide Web Conference, International World Wide Web Conferences Steering Committee, 2018, pp. 773–782..
[28] Julian McAuley, Jure Leskovec, Hidden factors and hidden topics: understanding rating dimensions with review text, in: Proceedings of the 7th ACM Conference on Recommender Systems, ACM, 2013, pp. 165–172.
[29] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, Jeff Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, 2013, pp. 3111–3119..
[30] Andriy Mnih, Ruslan R. Salakhutdinov, Probabilistic matrix factorization, in: Advances in Neural Information Processing Systems, 2008, pp. 1257–1264..
[31] Juan Ni, Zhenhua Huang, Jiujun Cheng, Shangce Gao, An effective recommendation model based on deep representation learning, Information Sciences 542 (2021) 324–342.
[32] Slava Novgorodov, Ido Guy, Guy Elad, Kira Radinsky, Generating product descriptions from user reviews, in: The World Wide Web Conference, 2019, pp. 1354–1364..
[33] Ashutosh Pandey, DeLiang Wang, Tcnn, Temporal convolutional neural network for real-time speech enhancement in the time domain, in: ICASSP 2019 – 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 6875–6879.
[34] Jeffrey Pennington, Richard Socher, Christopher Manning, Glove, Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.
[35] Julian Salazar, Katrin Kirchhoff, Zhiheng Huang, Self-attention networks for connectionist temporal classification in speech recognition, in: ICASSP 2019 – 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 7115–7119.
[36] Sungyong Seo, Jing Huang, Hao Yang, Yan Liu, Interpretable convolutional neural networks with dual local and global attention for review rating prediction, in: Proceedings of the Eleventh ACM Conference on Recommender Systems, ACM, 2017, pp. 297–305.

[37] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The Journal of Machine Learning Research 15 (1) (2014) 1929–1958.
[38] Yi Tay, Anh Tuan Luu, Siu Cheung Hui, Multi-pointer co-attention networks for recommendation, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Association for Computing Machinery, New York, NY, USA, 2018.
[39] Tijmen Tieleman, Geoffrey Hinton, Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, COURSERA: Neural Networks for Machine Learning 4 (2) (2012) 26–31.
[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008..
[41] Wu Libing, Cong Quan, Chenliang Li, Qian Wang, Bolong Zheng, Xiangyang Luo, A context-aware user-item representation learning for item recommendation, ACM Transactions on Information Systems (TOIS) 37 (2) (2019) 22.
[42] Jining Yan, Lin Mu, Lizhe Wang, Rajiv Ranjan, Albert Zomaya, Temporal convolutional networks for the advance prediction of enso, Scientific Reports 10 (2020) 8055..
[43] Jiaxuan You, Yichen Wang, Aditya Pal, Pong Eksombatchai, Chuck Rosenburg, Jure Leskovec, Hierarchical temporal convolutional networks for dynamic recommender systems, in: The World Wide Web Conference, ACM, 2019, pp. 2236–2246..
[44] Weihua Yuan, Hong Wang, Yu. Xiaomei, Nan Liu, Zhenghao Li, Attention-based context-aware sequential recommendation model, Information Sciences 510 (2020) 122–134.
[45] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, Shaoping Ma, Explicit factor models for explainable recommendation based on phrase-level sentiment analysis, in: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, ACM, 2014, pp. 83–92.
[46] Lei Zheng, Vahid Noroozi, Philip S Yu, 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, ACM, 2017, pp. 425–434.
[47] Xiaolin Zheng, Menghan Wang, Chaochao Chen, Yan Wang, Zhehao Cheng, Explore: Explainable item-tag co-recommendation, Information Sciences 474 (2019) 170–186.
[48] Chunting Zhou, Chonglin Sun, Zhiyuan Liu, Francis Lau, A c-lstm neural network for text classification, arXiv preprint arXiv:1511.08630, 2015..
[49] Cong Zou, Zhenzhong Chen, Joint latent factors and attributes to discover interpretable preferences in recommendation, Information Sciences 505 (2019) 498–512.