Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.051046

ARTICLE

Learning Dual-Layer User Representation for Enhanced Item Recommendation

Fuxi Zhu[1], **Jin Xie**[2,*] **and Mohammed Alshahrani**[3]

[1]Applied Research Center of Artificial Intelligence, Wuhan College, Wuhan, 430212, China

[2]College of Computer Science, South-Central MINZU University, Wuhan, 430074, China

[3]Unmanned. Company, Riyadh, 11564, Saudi Arabia

*Corresponding Author: Jin Xie. Email: jinxie@scuec.edu.cn

Received: 26 February 2024 Accepted: 23 May 2024 Published: 18 July 2024

ABSTRACT

User representation learning is crucial for capturing different user preferences, but it is also critical challenging because user intentions are latent and dispersed in complex and different patterns of user-generated data, and thus cannot be measured directly. Text-based data models can learn user representations by mining latent semantics, which is beneficial to enhancing the semantic function of user representations. However, these technologies only extract common features in historical records and cannot represent changes in user intentions. However, sequential feature can express the user's interests and intentions that change time by time. But the sequential recommendation results based on the user representation of the item lack the interpretability of preference factors. To address these issues, we propose in this paper a novel model with Dual-Layer User Representation, named DLUR, where the user's intention is learned based on two different layer representations. Specifically, the latent semantic layer adds an interactive layer based on Transformer to extract keywords and key sentences in the text and serve as a basis for interpretation. The sequence layer uses the Transformer model to encode the user's preference intention to clarify changes in the user's intention. Therefore, this dual-layer user mode is more comprehensive than a single text mode or sequence mode and can effectually improve the performance of recommendations. Our extensive experiments on five benchmark datasets demonstrate DLUR's performance over state-of-the-art recommendation models. In addition, DLUR's ability to explain recommendation results is also demonstrated through some specific cases.

KEYWORDS

User representation; latent semantic; sequential feature; interpretability

1 Introduction

User representation learning refers to building a user's interest representation through the analysis of user behavior and preferences, as well as the modeling and learning of user-related data. This is an important step towards a personalized recommendation system. A common research direction in user modeling is based on representation learning, using special machine learning algorithms to model or represent users or behaviors [1–4].

Among the numerous user-related data, text usually contains users detailed descriptions, evaluations, opinions and other information about items, which can provide a deeper understanding of user interests. Compared with simple click records or rating data, review can provide richer and more fine-grained user feedback. Moreover, review can provide certain contextual information to help understand the user's motivation and background. Finally, the review may contain some implicit interests and needs. Therefore, review can better capture users' interests and hobbies and provide more accurate, comprehensive and personalized recommendations.

However, using review for user representation learning and applying it to item recommendation also faces some challenges and difficulties: 1) Data sparsity: Review usually has high sparsity. This leads to data imbalance and sparsity problems when training representation models. 2) Complexity of semantic understanding: User review texts are usually subjective and individual differences, so there are complex semantic structures. Different users may have different reviews on the same item, and the reviews on different items may also be diverse. 3) Contextual understanding and time-effectiveness: User reviews are usually generated in specific contexts. Therefore, the acquisition and utilization of contextual information need to be considered. Additionally, users' interests and preferences may change over time.

After the emergence of transformer, applying pre-training and self-attention mechanisms to natural language processing can alleviate the complex semantic structure in review texts and deepen the semantic understanding of the context. However, in item recommendation, simply using transformer to extract text semantics is not complete enough, and does not take into account the role of the interactive relationship between items and users on key phrases in reviews. At the same time, the data sparsity problem in item recommendation, the time-effectiveness of user representation and recommendation explanation have not been solved.

Towards this end, we propose dual-layer user representation learner, named DLUR. This framework can utilize rating information as a supplement to review text data to alleviate the data sparsity problem. The semantic understanding layer adopts a semantic representation method from words to sentences and then transitions to paragraphs, extracts parts with strong semantic relevance in layers, and effectively parses complex semantic structures. In addition, in terms of the division of data paragraphs, paragraphs are collected from the two perspectives of users and items, maintaining the user's subjectivity and the diversity of item content. Finally, text feature extraction is used to process the context. The sequence feature layer extracts sequence features to solve the timeliness of user representation. The resulting representation learning framework is also capable of refining interpretable sentences.

Specifically, the core part of DLUR revolves around user interest representation learning. By integrating user interest representation and item representation, and using the pairwise learning method to train the model, the items that the user is interested in can be predicted. Moreover, in the process of user representation learning, we use user-item interaction, and the designed integration model can calculate the weight of text sentences, and the key sentence patterns extracted can be used as explanation sentences. User interest representation learning is divided into three parts: latent factor learning, text representation learning and sequential factor learning. In the first component, we utilize LFM to extract the long-term latent factors of users in ratings. In the second component, we add an interactive attention layer to the Transformer model to further increase the weight by integrating interactive attention and self-attention, improve the accuracy of semantic extraction, and thus mine the interesting parts of users' comment data. In the third component, we utilize item reviews to mine sequence factors to capture users' dynamic interest changes in a timely manner. To summarize our main contributions of this paper as follows:

1. This paper proposed a novel user representation learning method, called DLUR, that is capable of 1) having the ability to learn from sequences, and 2) capturing relationships between users and items. and, 3) extracts multi-form factors to ensure the versatility of user representation.
2. This paper applied DLUR in the recommendation process and can provide recommendation explanations at the same time.
3. This paper has extensive experiments on two public datasets demonstrated the superiority of DLUR compared to the recent state-of-the-art methods. A further appeal of DLUR is its applicability in real-world scenarios, which validates possibility of adopting DLUR on various Web platforms.

The remainder of the paper is organized as follow. In Section 2, we highlight the relevant works of recommender system and text representation. The framework and detailed construction of our model are introduced in Section 3, and Section 4 applies the model in recommender system. Section 5 presents the results and analysis of the experiments. Section 6 concludes the paper and provides suggestions for further research.

2 Related Work

2.1 Recommender Systems

Since the recommendation system lacks a certain understanding of the relationship between users and recommended items, in other words, it is indifferent to the interaction between users and items, resulting in a scarcity of data that can be used for recommendations. The main methods to solve the problem of data sparsity can be subdivided into context, collaborative filtering and algorithm-based improvement optimization.

Context-aware recommendations can alleviate the data sparsity problem. Jannach and Ludewig use different time divisions for evaluation to reduce the amount of data required for training and improve the efficiency of algorithm learning [5]. CoSeRNN is a neural network architecture that models a user preferences as a series of embeddings, one per session. By using approximate nearest neighbor search algorithm, context-sensitive instant recommendations are efficiently generated [6]. Unger et al. integrated contextual information into the neural collaborative filtering recommendation method and proposed three deep context-aware recommendation models based on explicit, unstructured and structured latent representations of contextual data [7]. Zheng et al. used multi-angle attribute interaction and local lifting technology to effectively capture different levels of interesting factors, improve the scoring effect, and also alleviate the problem of data sparsity [8].

As one of the most successful strategies in recommendation algorithms, collaborative filtering recommendation has a wide range of applications, such as Grouplens, Ringo, Tapestry and other commercial recommender systems. Collaborative filtering is traditionally divided into two categories: one is memory-based, which uses the entire user browsed and purchased product database to generate prediction results; the other is model-based, which builds a hierarchy model of user preferences before product recommendations. Gong et al. proposed to improve the structural similarity and numerical similarity respectively, and combined the two to obtain a user similarity calculation method that takes into account both structure and numerical value [9]. Zhang proposed a collaborative filtering recommendation algorithm based on user-item mixture model, which improves data sparsity by introducing user interest factors and item semantics [10]. Sun et al. used a pre-filling algorithm based on sentiment analysis to fill the sparse rating matrix to obtain a dense matrix [11].

There are also some data preprocessing strategies that lead to improved performance of recommendation algorithms on sparse data. For example, in [12], the user's interests are expressed as some topics through shallow semantic analysis, and a full probability formula is used to predict the topics of interest to the user. Mao et al. proposed a collaborative filtering algorithm based on Sigmoid function, which can effectively alleviate the problem of data sparseness and improve recommendation quality [13]. Poirson et al. proposed a method based on emotional evaluation. However, in practical applications, this strategy inevitably encounters difficulties in emotion perception and duration [14]. Ajoudanian et al. proposed a new fuzzy C-means clustering method. This method solves the sparsity problem by using the sparsest subgraph detection algorithm to define the initial center of the clustering method [15]. Although the above three methods can improve a certain recommendation effect, most of the data sources come from ratings. From the perspective of the development of recommendation systems, a single rating data source can mine limited user interests and cannot intuitively express user interests.

After the emergence of Transformer, many models use the composition principle of Transformer or the self-attention mechanism to build new models to complete recommendations based on temporal factors. As a method based on attention mechanism, SASRec takes into account both Markov chain and RNN-based methods. This model can capture long-term semantics while also targeting fewer actions using an attention mechanism [16]. DIEN designs an interest extraction layer to capture temporal interests from historical behavior sequences. In the evolutionary layer of interest, the attention mechanism is innovatively embedded into the sequential structure [17]. The BST model uses the Transformer model to capture the associated characteristics of each item in the user's historical sequence. And by adding the items to be recommended, the correlation with the items in the behavior sequence can be extracted [18]. RNN and its extension method GRU can model causal models in user sequences using nonlinear transitions between consecutive hidden states. Recommendation methods based on Transformer have many advantages. It can learn from variable-length inputs, learn from long-term dependencies, stimulate the vitality of sparse data, and compress hidden states. The shortcomings of this method are: complex structure and configuration, high hardware requirements, and lack of interpretability.

2.2 Text Representation Learning

In recent years, deep neural networks have become the main technology for user interest representation learning. Among the many deep structured semantic models (DSSM) [19], deep or neural factorization machines (DeepFM/NFM) [20,21] have become some representative works based on supervised representation learning.

Currently, in the field of natural language processing, a large amount of work has been focused on the direction of unsupervised models of sentence or paragraph vectors. The paragraph vector DBOW model is an unsupervised algorithm that learns fixed-length factor representations from variable-length text fragments [22]. Hill et al. proposed two new phrase or sentence representation learning goals: Sequential Denoising Autoencoding (SDAE) and FastSent, which is a sentence-level linear bag-of-words model [23]. A sentence embedding uses a latent variable generation model to provide a theoretical explanation of sentences in an unsupervised approach that can defeat complex supervised methods including RNN and LSTM [24]. These excellent models are independent and unordered based on single sentences. But in the actual context, there are many different forms of text expression, so all the sentences in the paragraph are not unrelated. Therefore, paragraph vectorization needs to take into account the order of sentences.

The emergence of attention mechanism research [25] simplifies the above problems. The attention mechanism is a technology that allows the model to focus on important information and fully learn and absorb it. It is not a complete model, but should be a technology that can be used in any sequence model. And another paper proposed by Google takes the idea of attention to the extreme. This paper proposes a brand-new model-Transformer [26], which abandons the CNN and RNN used in previous deep learning tasks. BERT [27] is built based on Transformer. This model is widely used in the NLP field, for example: Machine translation, question answering systems, text summarization and speech recognition, etc. The main innovations of the model are in the pre-training method, which uses two methods: occlusion language model and next sentence prediction to capture word-and sentence-level vector representations, respectively. It is this pre-trained language model that opens a new chapter in natural language processing.

In many natural language processing scenarios, there are relatively few supervised data, and the introduction of larger-scale unsupervised data can improve the effect. This is the main reason why BERT is widely popular in the field of natural language processing. In addition, language itself is normative, and this norm has great universality for different natural language processing tasks. Therefore, regular migration can be performed through BERT. However, in the recommendation field, there is a large amount of supervision data. The recommended users themselves do not have strong regularity, and they change rapidly. The rules are not universal and difficult to migrate. Moreover, BERT needs to make use of large-scale data to fully learn various knowledge such as semantics in the text through pre-training, and then use it for downstream tasks. Therefore, while BERT can bring better results for text-dependent recommendation scenarios, such as news recommendation, BERT is difficult to implement on low computing power devices. Moreover, there is a problem that the training process requires a large amount of unsupervised text data, which has low interpretability, and the model compression process leads to a performance loss of the language model on the inference task [28]. Therefore, this article does not directly use the BERT model, but improves it from the bottom layer of the transformer, making the newly obtained model more suitable for recommended data sources.

2.3 User Representation Learning

Due to the problem of data sparsity, a single text representation cannot fully represent user portraits. Industry experts have sought factors that affect user representation from many aspects and have proposed a variety of user representation learning methods.

TERACON introduces an embedding for each task, which is utilized to generate task-specific soft masks that not only allow the entire model parameters to be updated until the end of training sequence, but also facilitate the relationship between the tasks to be captured [29]. In DUVRec, a user preference is learned based on the representations of two distinct views, i.e., item view and factor view. Specifically, the item-view user representation is learned as the previous sequential recommendation, while the factor-view user representation is learned by a coarse-grained graph embedding method [30]. RobustSR with social regularization and multi-view contrastive learning, which aim to enhance the model awareness of relation informativeness and the discriminativeness of user representations [31]. RecGURU [32], JNET [33], LDBR [34] are learned models which can solve practical problems from the perspective of user representation and have achieved good experimental results. Therefore, this article is also inspired by the above model, and based on the extraction of original text factors, adds effective factor data and learns user representations.

3 The Proposed Model

We now present our item recommendation framework as follow figure. In Fig. 1, the core of this framework is user representation, and this part is mainly composed of semantic layer which include ratings factors and text representation extracted from review text, and user sequential representation. The representations of these two layers were integrated into user interest representations.

In terms of data collection for user interest representation, in addition to user comment texts, user latent factors extracted from ratings are also added. The data sources are more abundant, and the rating data is larger than the review data, which can make up for the lack of review data. Moreover, this method can also directly use the user's latent factors mined from the rating data as the user's long-term interests when the comment data was missing.

3.1 User Latent Factor

The representation of user's long-term latent factors adopts the LFM model. The rating matrix $R_{m,n}$ is expressed as the ratings of n items by m users, which is a quite sparse matrix. At the same time, $r_{i,j}$ represents the rating of item j by user i. In LFM, $R_{m,n}$ can be expressed as the product of two matrices. One is $P_{m,F}$ that each row of P represents the user's interest in each latent factor, and F represents the number of latent factors. The other matrix is $Q_{F,n}$, and each column represents the distribution of items on each latent factor. The following is the scoring formula for LFM:

$$\hat{r}_{i,j} = \sum_{f=1}^{F} P_{if} Q_{fj} \tag{1}$$

In order to prevent overfitting, a regular term is added to the objective function after control:

$$Loss_{min} = \sum_{r_{i,j} \neq 0} \left(r_{i,j} - \hat{r}_{i,j}\right)^2 + \lambda \left(\sum P_{if}^2 + \sum Q_{fj}^2\right) = f\left(P, Q\right) \tag{2}$$

The decomposed P and Q are the user latent factors and item latent factors required in the model structure diagram. In Fig. 1, the lower part of the semantic layer of the blue dotted box represents the user latent factor.

3.2 Text Latent Factor Extraction

In addition to ratings that can characterize users or items, user reviews are also a source of data that can intuitively express user interests. The entire text factor extraction process is shown in the figure below:

In the Fig. 2, the gray part is the prototype of Transformer. An interactive attention layer is added to the text factor extraction process. The technical idea is that through the interaction between users and item reviews, it is possible to further identify which words in the review sentences are key words in the user's personality expression. Integrating interactive attention and self-attention can lead to a more focused vector representation.

For word vectorization, the Sent2vec [35] unsupervised learning method is selected to create word vectors based on contextual information. The objective function is as follows:

$$\min_{U,V} \sum_{S \in C} \sum_{w_t \in S} \left(\ell\left(u_{w_t}^{\mathrm{T}} v_{S \setminus \{w_t\}}\right) + \sum_{w' \in N_{w_t}} \ell\left(-u_{w'}^{\mathrm{T}} v_{S \setminus \{w_t\}}\right) \right) \tag{3}$$

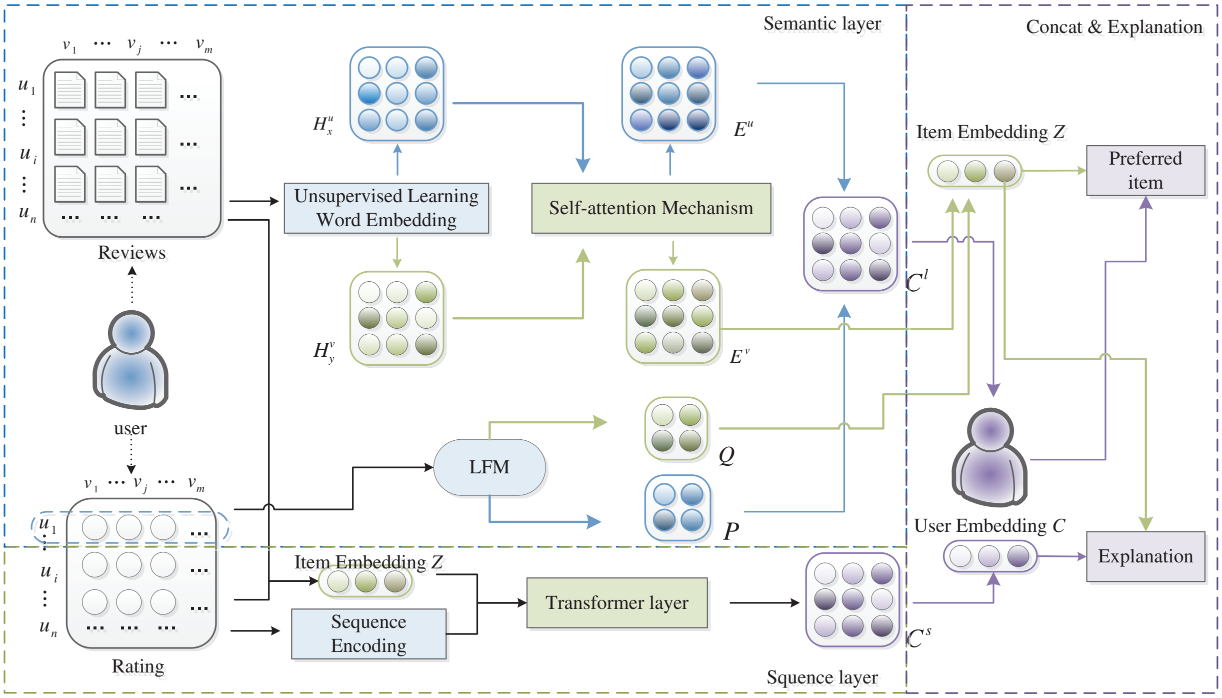

Figure 1: Illustration of the proposed dual-perspective embedding user representation learner (DLUR) for the explainable item recommendation

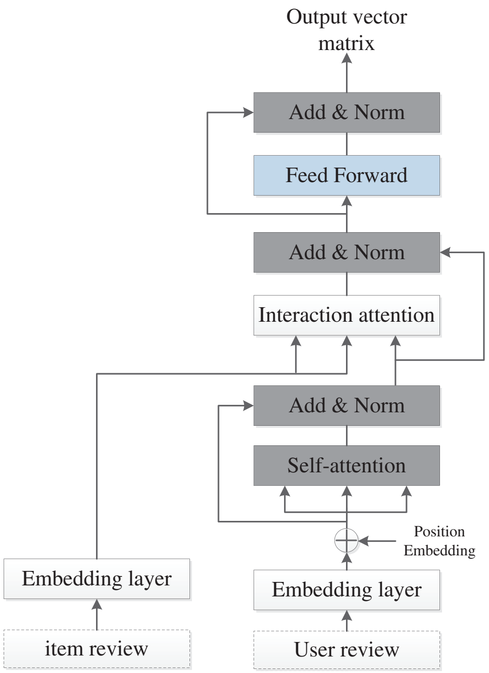

Figure 2: Word-level text factor extraction process

w_t represents the t-th word in sentence S, $w_t \in S$. N_{w_t} represents the negative sampling of the t-th word. u_w represents the target word vector. v_w represents the source word vector. Thus, the review documents of user x and item y are transformed into vector matrices: $H_x^u = \{w_0^u, w_1^u, w_2^u, \ldots w_n^u\}$, $H_x^u \in \mathbb{R}^{d\times n}$ and $H_y^v = \{w_0^v, w_1^v, w_2^v, \ldots w_m^v\}$, $H_y^v \in \mathbb{R}^{d\times m}$, where n and m represent the review lengths of user x and item y, respectively. Then a self-attention mechanism is further applied to words to capture long-distance dependencies in comments. It can be calculated as follows:

$$ATT\,(H) = softmax\left(\frac{\left(HW_r^Q\right)^T\left(HW_r^K\right)}{\sqrt{d_k}}\right)\left(HW_r^V\right) \tag{4}$$

Among them, W_r^Q, W_r^K and W_r^V are all learning parameters, and d_k is the dimension size. In addition, the self-attention mechanism in Transformer is implemented in parallel using g-heads, where each head calculates attention according to formula (4). The output of multi-head attention is the concatenation of g heads, followed by a linear mapping:

$$MultiHead\,(H) = f\left(G_1, G_2, \ldots G_g\right) W^M \tag{5}$$

$$G_i = ATT\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{6}$$

Among them, f is the join operation. W_i^Q, W_i^K, W_i^V are the corresponding query, key, and value weight matrices under each header, which are all learnable parameter matrices. Therefore, the user review vector matrix can be expressed as E_{ATT}^U. The item review vector matrix E_{ATT}^V can be obtained using the same process.

After the user review vector has been obtained, it is necessary to interact with the item review data to further highlight the influence of words. Inspired by [36,37], we use an attentive matrix $A^w \in \mathbb{R}^{d\times d}$ to derive a vector containing the importance of each word for both U and V. Specifically, the matrices U and V are mapped to the same latent space, and the correlation of each user-item pair is calculated as follows:

$$F_{i,j}^w = tanh\left(w_i^{uT} A^w w_j^v\right) \tag{7}$$

In the formula, w_i^u represents the factor vector of the i-th word in the review document of user x, $w_i^u \in E_{ATT}^U$. w_j^v represents the factor vector of the j-th word in the review document of item y. $F_{i,j}^w$ represents the correlation between w_i^u and w_j^v, where row $F_{i,*}^w$ contains the correlation between all word factor vectors in w_i^u and V_y. Similarly, column $F_{*,j}^w$ contains the correlation between the factor vectors of all words in w_j^v and U_x. The mean pooling operation of row F and column F is as follows:

$$g_i^{uw} = mean\left(F_{i,1}^w, \ldots, F_{i,m}^w\right) \tag{8}$$

According to the above correlation formula, the importance of the eigenvectors in D_x^u is highlighted.

$$a_i^{uw} = \frac{exp\left(g_i^{uw}\right)}{\sum_k^n exp\left(g_k^{uw}\right)} \tag{9}$$

a_i^{uw} represents $U_{i,*}$ attention weight at word granularity.

$$v_w = v_w a_i^{uw} \tag{10}$$

The resulting vector matrix is E^U_{inter}. Then use the residual network for normalization:

$$E^U_{word} = layerNorm\left(E^U_{ATT} + E^U_{inter}\right) \tag{11}$$

The result is a vector representation of user interests through the comment text. However, all resulting vectors are word-level. User interest representation requires the overall characteristics of the user, so it is necessary to integrate factor vectors with sentence semantics based on the factor vectors of words.

$$v_s = \frac{1}{|R(S)|}\sum_{w\in R(S)} v_w \tag{12}$$

Among them, $v_w \in E^U_{word}$. After obtaining the factor vector of the sentence, we can now consider the factor vector of the paragraph as a whole. The entire process is the word-level text factor extraction process in Fig. 2. Now the input that needs to be replaced is replaced by a sentence-level factor vector. After going through the entire process from formulas (4) to (11), the result is the sentence factor vector E^U_{sent}.

$$v_p = \frac{1}{|R(p)|}\sum_{w\in R(p)} v_s \tag{13}$$

Among them, $v_s \in E^U_{sent}$. The matrix composed of v_p is the final paragraph-level factor matrix E obtained from the text. The rows in the matrix represent the text factors of each user.

According to Fig. 1, in the right of the semantic layer, the user interest vector C^l representation should be the integration of the latent vector P and the text vector E.

$$C^l = P \oplus E \tag{14}$$

3.3 User Sequential Factor

Inspired by BST [18], user sequential factors are represented by factors extracted from item reviews using Transformer. The factor extraction process is shown in the Fig. 3.

In the Fig. 3, the Embedding Layer in the figure is mainly responsible for the conversion of item factor vectors and position factor vectors. The item factor extraction is shown in Fig. 1. It adopts an extraction process similar to the user interest vector C^l and integrates the text factors of the item and the latent factors of the item decomposed by LFM to obtain the item embedding Z. Positional embedding compares the value method of positional factors in BST. However, since the rating sequence is different from the click sequence, the interval time is uncertain. Compared with the click sequence in the session, the time interval will be larger and the reference is not great. Therefore, the one-hot hard coding method is directly used.

Scaled dot-product attention in Transformer is defined as follows:

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{15}$$

where Q represents the queries, K the keys and V the values. In our scenario, item embedding is taken as input, and they are converted into three matrices through linear projection and fed into the attention layer.

$$S = MH(C^s) = Concat(head_1, head_2, \ldots, head_h)\, W^H \tag{16}$$

$$head_i = Attention\left(EW^Q, EW^K, EW^V\right) \tag{17}$$

where the projection matrices, W^Q, W^K, $W^V \in R^{d\times d}$ and C^s is the user's sequential factor matrix output after passing through the Transformer layer.

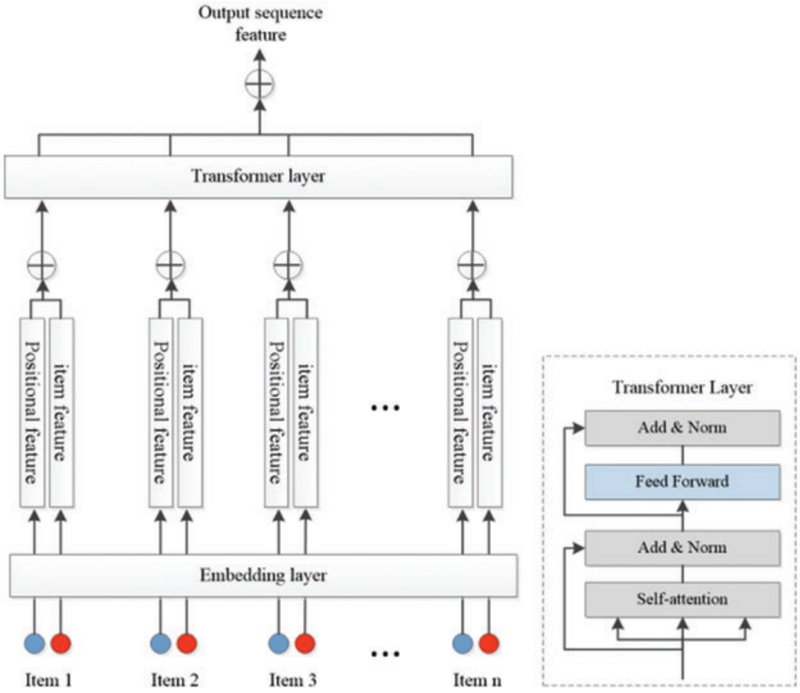

Figure 3: Sequential factor extraction process

Based on the obtained user interest factors and user sequential factors, a relatively complete user portrait factor C can be obtained.

$$C = concat\left(C^l, C^s\right) \tag{18}$$

4 Application of User Representation in Recommendation

The design of the recommendation method is based on the idea of user representation. Among the available data resources, the user's latent features are decomposed using ratings as part of the long-term interests, and are integrated with the long-term user interest features extracted from the text to form a feature that can richly express the user's consistent interests. Interest changes extracted through time series can also express the user's current status. Combining the two can fully express user information. Given a user u, the goal of the paper is to construct a user interest representation based on the user's multi-dimensional factor, and then compare it with candidate items to recommend items with high similarity. At the same time, the recommendation structure will provide a sentence-level explanation mechanism.

After obtaining the user representation C, the features of the items are extracted in the same way. As can be seen from Fig. 1, the extraction process of item feature Z is consistent with the extraction method of user long-term representation. $Z = [Z_{ATT}; Q]$ integrates the feature Z_{ATT} in the item review data and the latent feature Q of the item decomposed from the rating matrix. From this idea, the following scores can be calculated:

$$score_{x,y} = \varphi\left(C_x, Z_y\right) \tag{19}$$

The pairwise learning method was selected to train the model. All user-selected items with ratings and reviews are used as positive samples. Randomly select the next item from other sessions in the same batch as a negative sample. These positive and negative samples are used to train the entire neural network. The BPR loss function in the pairwise method applied to the personalized recommendation system is adopted:

$$Loss = -\frac{1}{N} \cdot \sum_{j}^{N} \log\left(\sigma\left(s_{x,i} - s_{x,j}\right)\right) + \lambda\left(||\theta||^2\right) \tag{20}$$

Among them, N is the number of negative sampling samples. $s_{x,i}$ is the positive sampling sample score. $s_{x,j}$ is the negative sampling sample score. σ is the sigmoid function. λ is the l_2 regularization hyperparameter. θ represents the parameters of the model.

Depending on the $score_{x,y}$, a list of items can be recommended to the user x. This list is also the result of the application of user interest representation in recommendations. The recommendation results can include not only the user's textual semantic features, but also the latent features in the ratings, and also include the user's temporary change features. The accuracy of the recommended item list is relatively high.

The method based on user representation proposed in this article can also solve the problem of recommendation explanation. Many interpretations of object-based features or aspects are based on words or phrases, but this method is prone to cause semantic ambiguity or incomplete expression. If all reviews of recommended items are used as an explanation, there will be redundancy. After all, there are many sentences in the reviews of items, but users may only pay attention to part of them. Therefore, review sentences that can be used for explanation become the key to setting up the explanation mechanism. In Section 3.2, the model uses the nature of the interaction between the user and the item and the attention mechanism to successfully find the sentence with high attention among the many reviews of the user on an item, so the sentence can be used as a recommended explanation.

The sentence-level comment feature vector v_s^u of user x has been obtained by formula (12), and the sentence-level comment feature vector v_s^v of item y can also be obtained by formula (12). According to the process of extracting text features in Fig. 2, after going through the process from formulas (4) to (8), we can get:

$$a_j^{vs} = \frac{exp\left(g_j^{vs}\right)}{\sum_k^m exp\left(g_k^{vs}\right)} \tag{21}$$

a_j^{vs} is the weight of the item in the j-th sentence at the sentence level. Moreover, this weight is obtained after the user interacts with the item. The higher the value, the greater the influence of the sentence on the user. From this, sentences with higher weights can be selected as recommended explanations.

5 Experiments

In the experimental part, multiple experiments were designed to verify the overall performance of the model and the technical advantages of each part. First, three recommendation indicators of the recommendation system are used to compare with the baseline to verify the recommendation effect that the characterization model can achieve. Then the effectiveness of each part of the features that make up the user representation is verified separately to demonstrate the advantages of the model. Finally, the visualization of the text weight and the selected high-weight sentences are used to generate recommended explanation sentences.

5.1 Experimental Setup

The experimental part uses four popular data sets from Amazon and the Yelp dataset for experiments. The four data sets are: "Cell Phones and Accessories", "Clothing Shoes and Jewelry", "Electronics" and "Toys and Games". Each data set contains "user ID", "product ID", "rating", and "review text". Meanwhile, we chose reviews from Yelp in 2019. The experiment selected items that contained reviews. Then the user's interest factor is extracted from the review.The basic statistics of the datasets are shown in Table 1.

Table 1: Data set feature statistics

Datasets	#Users	#Products	#Review	Avg review length	Density
Phones	3216	9018	47139	135.39	0.1625%
Clothing	5200	20424	72142	35.88	0.0679%
Electronics	45225	61918	773502	160.83	0.0276%
Toys	4188	11526	74423	103.09	0.1541%
Yelp_2019	545241	128228	1215836	97.53	0.0017%

Data preprocessing for the data set:

(1) The text is divided into different documents based on userID and itemID. Each user's review of an item acts as a paragraph in the document.
(2) Each paragraph is divided by punctuation marks, with one sentence per line.
(3) All letters in each sentence are converted to lowercase letters.
(4) Use Natural Language Toolkit (NLTK) to complete word segmentation. In addition, the data set is filtered so that each user has at least 10 or more item options, regardless of whether there is comment data, and the rest are deleted.

Divide each data set into three groups: training set, validation set, and test set. For each data set, the last item record selected by the user is retained as the test set, the penultimate selected record is used as the validation set, and the rest is the training set. The experiment uses the training set to train the model, the validation set to adjust parameters, and finally the optimal parameter settings are applied to the test set to achieve the final recommendation result.

The hyperparameters of the comprehensive recommendation method are adjusted on the validation set. Set the number of heads h in the multi-head attention mechanism in the word-level and sentence-level text feature extraction process to 4. The entire text feature extraction includes the number of Transformer layers set to 6. Dimension size is 512 (adjusted in [128, 256, 512, 1024]). The dimension of the feedforward network is 2048. The dimensions of the word vector and the dimensions set by userID and itemID are all 300 (adjusted in [200, 300, 400]). To avoid transition fitting loss rate is set to 0.3 (adjusted in [0.1, 0.3, 0.5, 0.7]). Set the batch size to 400. The number of negative samples used is 5 for each positive sample. All parameters in the baseline model were adjusted with reference to the setting strategy in the original paper to adjust the hyperparameters in all methods.

The evaluation of experiments adopts common recommended standards, including HR (Hit Ratio), MRR (Mean Reciprocal Rank) and NDCG (Normalized Discounted Cumulative Gain). And generate a Top-10 item recommendation list for each user to observe the performance of the recommendation method.

- HR can be used to determine whether the correct items are included in the final recommended Top-20.

$$HR@K = \frac{NumberOfHits@K}{GT} \tag{22}$$

Among them, the denominator is all test sets, and the numerator is the number of test sets in the Top-k list.

- MRR is the average reciprocal ranking of desired items. This evaluation metric focuses on whether recommended items are placed in a higher position.

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \tag{23}$$

Among them, $rank_i$ is the ranking of the ith recommended item in the recommendation list.

- NDCG is widely used to measure sorting accuracy. If the item selected by the user is ranked higher in the recommendation list, the score is higher. What is used here is the average value of all users NDCG.

$$NDCG@K = \frac{\sum_{u\in U} NDCG_u@K}{IDCG_u} \tag{24}$$

5.2 Model Comparisons

To verify our DLUR's advantage, we evaluated DLUR's performance through the comparisons with the following baseline models and the ablated variants of our model.

The following baselines are the representative item models. Each model is similar to DLUR in terms of recommendation ideas or feature extraction ideas, but the technical routes are different, which better reflects the advantages of DLUR.

- DeepCoNN model [38]. This model simultaneously utilizes the semantic information in user and item reviews to construct their respective features.
- APSE [39] is a rating model that extracts user and item features by using reviews, and combines existing rating features to predict the ratings of unrated items. This method also uses scoring and attention mechanisms to extract user interest features. Compared with the recommendation method in this article, user features are extracted from text.
- PRSL is the result of previous research. The overall architecture of this method is similar to the recommendation method in this article. They both extract user interest features based on historical records and dynamic short-term changes.

- GRU4REC is a GRU-based serialization prediction model [40]. This model uses data sequences to extract sequence features of users within a short period of time. It is consistent with the idea of extracting some user interest features in the recommendation method of this article, which are all derived from sequence features.
- The AttRec model also make uses of the user's short-term and long-term interest characteristics to build a recommendation model [41]. The self-attention structure is used in short-term interest feature extraction, while long-term interest feature extraction comes from rating data. The idea of this model is similar to the recommendation framework of this article, but the technology used and data sources are different.
- SSG design a three-way encoder architecture that jointly captures long-term (set), short-term (sequence), and collaborative (graph) features of users and items for recommendation [42]. The common point between the model and the DLUR model is that they both use review and sequence features, but their implementation methods are different, and the methods of feature fusion are also different.The performance comparison of each model is shown in Table 2.

Table 2: Performance comparisons of all models on the five datasets

Dataset	Metric	DeepCoNN	ASPE	GRU4REC	AttRec	PRSL	SSG	DLUR
Phones	HR	0.1197	0.1612	0.1813	0.2037	0.2128	0.2232	0.2253
	MRR	0.0132	0.0301	0.0428	0.0435	0.0481	0.0484	0.0492
	NDCG	0.0447	0.0687	0.0711	0.0933	0.1032	0.1089	0.1097
Clothing	HR	0.0312	0.0494	0.0583	0.0712	0.0789	0.0805	0.0821
	MRR	0.0095	0.0123	0.0131	0.0139	0.0191	0.0218	0.0221
	NDCG	0.0164	0.0212	0.0315	0.0428	0.0536	0.0603	0.0613
Electronics	HR	0.0573	0.0817	0.0906	0.1036	0.1247	0.1402	0.1372
	MRR	0.0102	0.0194	0.0204	0.0215	0.0308	0.0420	0.0413
	NDCG	0.0287	0.0402	0.0490	0.0638	0.0715	0.0810	0.0805
Toys	HR	0.1284	0.1591	0.1703	0.2014	0.2217	0.2219	0.2311
	MRR	0.0178	0.0289	0.0322	0.0397	0.0427	0.0496	0.0510
	NDCG	0.0463	0.0692	0.0785	0.0951	0.1012	0.1079	0.1098
Yelp_2019	HR	0.0267	0.0293	0.0311	0.0320	0.0342	0.0361	0.0354
	MRR	0.0082	0.0091	0.0097	0.0114	0.0121	0.0205	0.0190
	NDCG	0.0197	0.0211	0.0214	0.0219	0.0227	0.0301	0.0295

Among the selected comparison models, there are score prediction models DeepCoNN and APSE. In the comparative experiment, items with high predicted scores are used as recommended items, and then compared according to HR, MRR and NDCG standards. GRU4REC is a recommendation model that uses temporal features. Compared with DLUR, it only considers changes in user interests in a short period of time. Current research shows that changes in user interests in a short period of time have a greater impact on the user's next item selection. At the same time, you can see that the end user's choice of items is still affected by the interest in historical record extraction. Therefore, from the perspective of comparative performance, the method in this article is still relatively good. Compared with three models, AttRec only uses rating data to extract long-term and short-term user interest features, PRSL only uses review data to extract user interest features, DLUR integrates latent features in ratings and reviews as well as semantic extraction features to make user portraits completer and more recommended. Our results are better than them. Compared with SSG, the recommendation system based on DLUR performs slightly lower on the datasets Yelp_2019 and Electronics. The main reason is that these two data sets have a large number of users and items, and there is a high proportion of interactions, but the proportion of reviews is low. Therefore, the interaction reflected by SSG's graph is better than the features extracted by review. Moreover, the three selected recommendation indicators are all used to measure the accuracy of recommendation ranking. In practical applications, DLUR can not only show that the recommendation list has high accuracy, but also shows a good advantage in ranking.

5.3 Ablation Experiment

Beside above item recommender baselines, we further compared following ablated variants:

- DLUR-factor: It only has the rating-view module. In other words, C^l is directly used as final user representation C to compute $score_{x,y}$ by formula (19).
- DLUR-sr: It only has the item-view module. In other words, C^s is directly used as final user representation C to compute $score_{x,y}$ by formula (19).
- PRL: The long-term interest expression part of PRSL is PRL, which uses user-item pair interaction to extract text features.
- PRS: The short-term interest representation part of PRSL is PRS, which uses GRU to extract short-term user interest features.
- DLUR-lfm: This model removes the LFM used for rating from DLUR. It uses reviews and sequence features to make recommendations.
- DLUR-re: This model removes the reviews factors from DLUR. It uses rating and sequence features to make recommendation.DLUR-att: This model removes the attention layer from DLUR. It uses word2vec to vector reviews.

It can be seen from formula (14) that part of the model fuses the latent vector P and the text vector E as user interest features. In the baseline, DeepCoNN and APSE are also recommendation systems completed using such a technical route. The experiments in this section will compare part of the features in the model with other similar technical routes, including the traditional word2vec encoding method and PRL, to demonstrate DLUR's technological improvements.The specific comparison results are shown in Table 3.

Table 3: Long-term interest feature performance comparison

Dataset	Metric	word2vec	PRL	DLUR-factor
Phones	HR	0.1643	0.1865	0.1937
	MRR	0.0205	0.0328	0.0415
	NDCG	0.0511	0.0798	0.0931
Clothing	HR	0.0412	0.0534	0.0720
	MRR	0.0117	0.0154	0.0175
	NDCG	0.0318	0.0376	0.0427
Electronics	HR	0.0634	0.0930	0.1022
	MRR	0.0108	0.0156	0.0194
	NDCG	0.0421	0.0580	0.0633

(Continued)

Table 3 **(continued)**

Dataset	Metric	word2vec	PRL	DLUR-factor
Toys	HR	0.1143	0.1540	0.1892
	MRR	0.0216	0.0350	0.0417
	NDCG	0.0475	0.0700	0.0891
Yelp_2019	HR	0.0112	0.0193	0.0231
	MRR	0.0049	0.0061	0.0084
	NDCG	0.0076	0.0093	0.0116

The experiment in this section only extracts features from text and ratings as user interest features for personalized recommendations. From this, we compare the effects of various text feature extractions. Although traditional word2vec is the most commonly used method of text vectorization in personalized recommendations. However, the recommendation effect is not as good as the semantic extraction effect of PRL. The recommendation method in this paper integrates text features and latent feature vectors in ratings, which shows that the feature vectors in ratings are also very helpful in improving the recommendation effect.

User temporal features will be compared with PRS. PRS utilizes semantic coding in the coding part, so the obtained temporal features also contain semantic information. However, DLUR takes into account the scarcity of user review data, and the recommendation method is a task-independent general method. Therefore, the form of ID encoding is used instead. The comparison results are shown in Fig. 4. In Fig. 4, method is the DLUR-sr model.

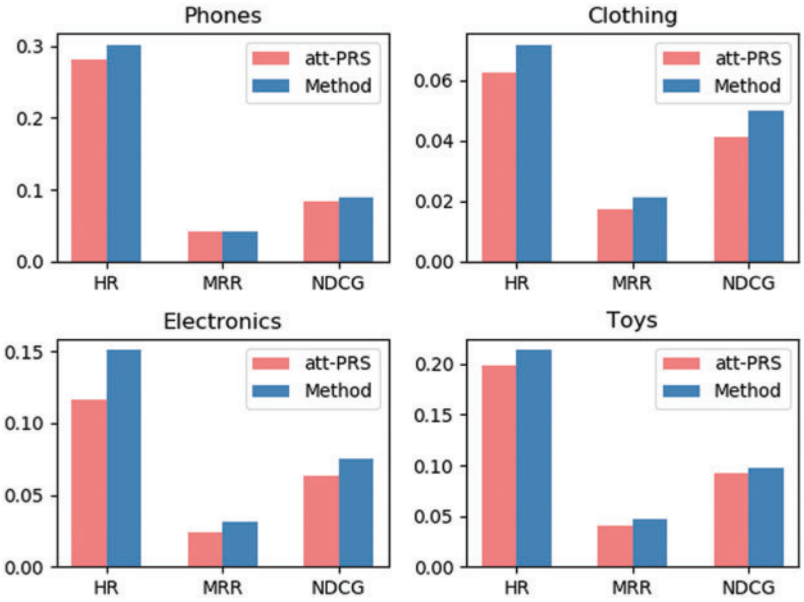

Figure 4: Comparison of time sequence features

As shown in the Fig. 4, among the four data sets, Clothing and Electronics, as two data sets with relatively low review density, have slightly improved in the comparison indicators. The other two data sets have similar comment densities. It can be seen from the figure that the performance of PRS and the method in this chapter are comparable. Comparing the technical ideas of the two methods, when there is sufficient review data, PRS performs better because it takes advantage of semantic features. When the review data is sparse, the advantages of recommendation methods based on user interests are very obvious.

DLUR mainly comes from two hierarchical structures, in which the attention layer is used. In order to reflect the completeness of the model, each part was removed separately in the ablation experiment to test the effect of the model. The comparison results are shown in Table 4.

Table 4: Comparison of the effects of various parts of DLUR

Dataset	Metric	DLUR	DLUR-lfm	DLUR-factor	DLUR-re	DLUR-att
Phones	HR	0.2253	0.1845	0.1937	0.2038	0.2143
	MRR	0.0492	0.401	0.0415	0.0473	0.0489
	NDCG	0.1097	0.0884	0.0931	0.0983	0.1078
Clothing	HR	0.0821	0.0701	0.0720	0.0793	0.0803
	MRR	0.0221	0.0168	0.0175	0.0204	0.0214
	NDCG	0.0613	0.0409	0.0427	0.0595	0.0606
Electronics	HR	0.1372	0.1005	0.1022	0.1294	0.1317
	MRR	0.0413	0.0178	0.0194	0.0402	0.0409
	NDCG	0.0805	0.0534	0.0633	0.0791	0.0800
Toys	HR	0.2311	0.1734	0.1892	0.2207	0.2243
	MRR	0.0510	0.0392	0.0417	0.0490	0.0503
	NDCG	0.1098	0.0790	0.0891	0.1052	0.1073
Yelp_2019	HR	0.0354	0.0209	0.0231	0.0321	0.0341
	MRR	0.0190	0.0073	0.0084	0.0175	0.0184
	NDCG	0.0295	0.0102	0.0116	0.0278	0.0283

For the data sets Yelp_2019 and Electronics with a small proportion of reviews, the DLUR-lfm model recommendation effect is relatively poor. The results reflected by other data sets are not ideal. The main reason is that the proportion of reviews is relatively small. Once the decomposition of ratings is missing, there will be fewer data features that can be mined and the recommendation effect will be compromised. For data sets with fewer reviews, the recommendation effect of DLUR-re is less affected. The three index values of the data set with more comments are quite different. And the model is unable to generate recommended explanations. DLUR-att removes the attention layer, and using only word2vec in the review part cannot establish user-item semantic interaction, and cannot accurately extract keywords with high attention. This will make non-keywords also have an impact on the extraction of user features and make it impossible to generate explanations. From Table 4, although the three indicator values of the DLUR-att model are better than DLUR-re, but they still cannot reach the indicator values of DLUR.

5.4 *Text Weight Visualization*

The important feature extraction of DLUR comes from text, and the text processing process includes two processes from word vectorization to sentence vectorization. An interactive attentive layer is added based on the transformer to focus more on the weight of the vector unit. The experiments in this section display important words and sentences in text paragraphs from a visual perspective. Therefore, we extracted a set of user-item pairs from a review in the dataset and visualized it. Table 5 shows all the comments corresponding to a specific user ID and uses various colors to show the influences of the sentences on the paragraph. We only show the top 5 sentences in the entire review in terms of importance.

Table 5: High weights sentences in the special user review documents

UserID: A2XU46XXNV19C8
I keep this board on top of the hallway table so that I can quickly write notes (which will not get lost until I erase them) and it folds down neatly so it is easy to hide when company calls. I also like the size–not too big and not too small. Quality magnets hold pretty good but I use it mainly for notes. NOTE: Children’s alphabet letters don’t hold very well–they tend to slide.
The bus arrived without the stop sign. In fact, it was packaged without the sign at all. It had clearly broken off but had been shipped to me anyway. Now, being it is going to a 4-yr-old boy, I never anticipated the stop sign to last long but it would have been nice to present it to my son with the sign still in tact. Not worth returning it for a replacement. Hopefully the rest of the bus will last a little bit longer.
This toy is truly one of those toys that will be sought after in 30 years when parents are looking for quality toys for their grandchildren–SO BUY IT NOW! This is a 5 star toy–no doubts about it. This toy is the ideal cause & effect toy. Babies and toddlers love the fun music, they are awed by the speed at which the balls fly out from the tube and they will eventually learn how to press the large button by themselves, and they are absolutely intrigued by the unpredictability of how the balls fly up and roll down the tube. FASCINATING!!! The balls do fly all over the place so just know that, but that is all part of the fun. Yes, you will definitely be looking for balls under your coffee tables, etc., but so what? This is such a great toy. THIS IS ALSO A GREAT TOY FOR CHILDREN WITH AUTISM.
This is really a simple, easy to use, and fun little toy. I’d call it a classic toy.
In my humble opinion, Vtech is \”the BEST\” educational toy company out there. We have never been let down by the quality, thoughtfulness and talent that goes into making Vtech toys. Both my children, one autistic (age 4) and the other not (age 2) are learning and having fun every time they turn on any of their creative learning devices. Another great company for electronic learning toys is LeapFrog–also very nice products.
Bought this for my 4-yr-old with Autism to help him with motion and movement. So far he is only interested in the sounds and is a bit timid to actually sit on it. However, I think as he gets older he will enjoy it more. Honestly though, wished I had checked Craig’s list first because this thing is extremely sturdy and will most likely hold up in good to excellent condition by the time your child has out grown it. Also, it would have already been put together for you by someone else! Lovely toy though–a true classic.

(Continued)

Table 5 **(continued)**

UserID: A2XU46XXNV19C8
Parents know that finding learning toys ideal for autism AND neurotypical siblings is a challenge. I have a 4-yr-old (autistic) and a 2-yr-old (neurotypical) and THEY BOTH LOVE THIS LEARNING TOY!!!! You could buy 2 if you don't want arguments but I bought one so they BOTH learn how to share and take turns. Wonderful job to the LeapFrog game creators, and thanks.

The sentences in Table 5 use three colors: red, orange, and yellow to indicate the sentence weight from high to low. It can be seen from the sentences with high weight in user comments that the user is most concerned about whether the toy is suitable for his two children. One child has autism, and the user repeatedly emphasizes the suitability of the toy for the child. In addition, users are more concerned about the functions of toys. It can also be seen that DLUR's refinement of text semantics is from the perspective of user-item pairs, and it can extract the points that users are most concerned about in an interactive way. This is a representative user characteristic.

In Table 6, we show the top 25% of the weighted words. It can be seen from these shading words that words with character identification have a high weight, such as "autistic" and "4-yr-old". In addition, some nouns have relatively high weight, such as "motion" and "learning". Through these we can also see what users focus on when choosing toys.

Table 6: Highlighted words in the top 3 sentences according to the attentive weights in the user review documents

UserID: A2XU46XXNV19C8
Both my children, one autistic (age 4) and the other not (age 2) are learning and having fun every time they turn on any of their creative learning devices.
I have a 4-yr-old (autistic) and a 2-yr-old (neurotypical) and THEY BOTH LOVE THIS LEARNING TOY!
Bought this for my 4-yr-old with Autism to help him with motion and movement.

5.5 Sentence-Level Explanation

In the explanation mechanism generation stage, we send recommended items and explanations to users at the same time. DLUR focuses on the generation of user portraits. After being applied to the recommendation system, the generated recommendation explanations are compared with the comments in the original test set, as shown in Table 7.

In Table 7, we selected different user evaluations of the same item as the real evaluation. After DLUR is applied to the recommendation system, predicted scores and recommendation basis are generated. There is not much difference between the rating and the actual value. The two reference samples were chosen to demonstrate whether the recommendation explanations generated by high-scoring and low-scoring evaluations on the same item can support the user's intention to choose toys. In Table 7, the high-scoring evaluation focuses on the sound, firmness, and movement of the toy horse, and the recommended explanation covers these aspects. In low-scoring evaluations, version and security are considered, and the recommended explanations also indicate the problem of version inconsistency. Overall, the explanation mechanism provided in DLUR meets user needs.

Table 7: Comparison between recommended explanation and real reviews

Pair$_{real}$ (user$_1$, item$_1$, 5.0)	Pair$_{prediction}$ (user$_1$, item$_1$, 4.876)
Review	Explanation
Bought this for my 4-yr-old with Autism to help him with motion and movement. So far he is only interested in the sounds and is a bit timid to actually sit on it. However, I think as he gets older he will enjoy it more. Honestly though, wished I had checked Craig's list first because this thing is extremely sturdy and will most likely hold up in good to excellent condition by the time your child has out grown it. Also, it would have already been put together for you by someone else! Lovely toy though-a true classic	We still have this horsie, still love it, and it is in great condition after 4 years. Very durable and handles well on a vigorous bouncy ride. The sound is great, not intrusive, and it really wears out the little ones with a high energy level!
Pair$_{real}$ (user$_2$, item$_1$, 3.0)	Pair$_{prediction}$ (user$_2$, item$_1$, 2.79)
Review	Explanation
I did not realize there had been a change in this horse until just now as I was going to review the one we have. I must admit the other one sounds a lot better in terms of features and one major important factor: safety. I am giving this horse 3 stars for that reason. The old one with stabilizers, yes, this one, no, which is a real shame because this toy would be perfect but for that important missing part.	The only thing I would change would be that the metal support rails had some kind of padding. Makes me nervous but not them. I would like to warn all of you reading these glowing reviews that most of the reviewers have the older version with many nice features. I hope buyers can track down the version we have!

6 Conclusions

In this paper, we focus on analyzing how users are represented and mining user hidden features from existing user reviews and ratings. In the process of mining latent features of text, the transformer-based model adds an interaction layer. So that the user's text representation content covers the characteristic information of the selected item. The advantage is that it can highlight the weight of keywords in sentences or paragraphs. The expression of user characteristics is more focused. On this basis, adding latent features decomposed by ratings can alleviate the lack of representation caused by the sparsity of review data. These historical comments and ratings can only reflect the general characteristics of users' choices over time, but cannot represent the changes in users' interests over time. Therefore, this article proposed a mining method of time series features. It takes fusion of user latent features and temporal features as user representation. From the experimental point of view, the results are very good.

The source data of user representation also has diverse content, including video, audio, etc. Future research will expand the data sources of user representation and mine user characteristics from videos, audios and images. Currently, there are many excellent models that can achieve user representation from a single data source, but the fusion method is relatively simple and cannot guarantee the commonality and diversity of user interests at the same time. Later research work will focus more on the study of integration methods.

Acknowledgement: The authors would like to thank the reviewers for their helpful suggestions which have considerably improved the quality of the manuscript.

Funding Statement: This research is supported by the Applied Research Center of Artificial Intelligence, Wuhan College (Grant Number X2020113) and the Wuhan College Research Project (Grant Number KYZ202009).

Author Contributions: The authors confirm their contribution to the paper as follows: study conception and design: Fuxi Zhu and Jin Xie; data collection: Mohammed Alshahrani; analysis and interpretation of results: Fuxi Zhu, Jin Xie and Mohammed Alshahrani; draft manuscript preparation: Fuxi Zhu and Jin Xie. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data supporting the results of this study are public datasets that can be directly searched on the Internet.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Y. Ni *et al.*, "Perceive your users in depth: Learning universal user representations from multiple e-commerce tasks," in *Proc. 24th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, London, UK, 2018, pp. 596–605. doi: 10.1145/3219819.3219828.

[2] J. Wang, F. Yuan, J. Chen, Q. Wu, M. Yang and Y. Sun, "StackRec: Efficient training of very deep sequential recommender models by iterative stacking," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, Canada, 2021, pp. 357–366. doi: 10.1145/3404835.3462890.

[3] F. Yuan, A. Karatzoglou, I. Arapakis, and J. Jose, "A simple convolutional generative network for next item recommendation," in *Proc. Twelfth ACM Int. Conf. Web Search Data Mining*, Melbourne, VIC, Australia, 2019, pp. 582–590. doi: 10.1145/3289600.3290975.

[4] K. Zhou *et al.*, "S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Ireland, 2020, pp. 1893–1902. doi: 10.1145/3340531.3411954.

[5] D. Jannach and M. Ludewig, "When recurrent neural networks meet the neighborhood for session-based recommendation," in *Proc. Eleventh ACM Conf. Recommender Syst.*, New York, NY, USA, 2017, pp. 306–310. doi: 10.1145/3109859.3109872.

[6] C. Hansen *et al.*, "Contextual and sequential user embeddings for large-scale music recommendation," in *Proc. 14th ACM Conf. Recommender Syst.*, Brazil, 2020, pp. 53–62. doi: 10.1145/3383313.3412248.

[7] M. Unger, A. Tuzhilin, and A. Livne, "Context-aware recommendations based on deep learning frameworks," *ACM Trans. Manag. Inf. Syst. (TMIS)*, vol. 11, no. 2, pp. 1–15, 2020. doi: 10.1145/3386243.

[8] L. Zheng, F. Zhu, and X. Yao, "Recommendation rating prediction based on attribute boosting with partial sampling," *Chin. J. Comput.*, vol. 39, no. 8, pp. 1501–1514, 2016. doi: 10.11897/SP.J.1016.2016.01501.

[9] L. Gong and J. Wang, "Research on collaborative filtering recommendation algorithm for improving user similarity calculation," in *Proc. 2021 1st Int. Conf. Control Intell. Robot.*, Guangzhou, China, 2021, pp. 331–336. doi: 10.1145/3473714.3473772.

[10] S. Zhang, "Research on recommendation algorithm based on collaborative filtering," in *2021 2nd Int. Conf. Artif. Intell. Inf. Syst.*, Chongqing, China, 2021, pp. 1–4. doi: 10.1145/3469213.3470399.

[11] P. Sun, J. Li, and G. Li, "Research on collaborative filtering recommendation algorithm based on sentiment analysis and topic model," in *Proc. 4th Int. Conf. Big Data Computing*, Guangzhou, China, 2019, pp. 169–178. doi: 10.1145/3335484.3335536.

[12] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009. doi: 10.1109/MC.2009.263.

[13] Y. Mao, J. Liu, H. U. Rong, M. Tang, and M. Shi, "Sigmoid function-based web service collaborative filtering recommendation algorithm," *J. Front. Comput. Sci. Technol.*, vol. 11, no. 2, pp. 314–322, 2017. doi: 10.3778/j.issn.1673-9418.1511072.

[14] E. Poirson and C. Da Cunha, "A recommender approach based on customer emotions," *Expert. Syst. Appl.*, vol. 122, no. 1, pp. 281–288, 2019. doi: 10.1016/j.eswa.2018.12.035.

[15] S. Ajoudanian and M. N. Abadeh, "Recommending human resources to project leaders using a collaborative filtering-based recommender system: Case study of gitHub," *IET Softw.*, vol. 13, no. 5, pp. 379–385, 2019. doi: 10.1049/iet-sen.2018.5261.

[16] W. C. Kang and J. Mcauley, "Self-attentive sequential recommendation," in *IEEE Int. Conf. Data Min.(ICDM)*, Singapore, 2018, pp. 197–206. doi: 10.1109/ICDM.2018.00035.

[17] G. Zhou *et al.*, "Deep interest evolution network for click-through rate prediction," in *Proc. AAAI Conf. on Artif. Intell.*, Hawaii, USA, 2019, pp. 5941–5948. doi: 10.1609/aaai.v33i01.33015941.

[18] Q. Chen, H. Zhao, W. Li, P. Huang, and W. Qu, "Behavior sequence transformer for e-commerce recommendation in Alibaba," in *Proc. 1st Int. Workshop on Deep Learn. Pract. High-Dimens. Sparse Data*, Anchorage Alaska, USA, 2019, pp. 1–4. doi: 10.1145/3326937.3341261.

[19] P. S. Huang, X. He, J. Gao, L. Deng, A. Acero and L. Heck, "Learning deep structured semantic models for web search using clickthrough data," in *Proc. 22nd ACM Int. Conf. on Inf. Knowl. Manag.*, San Francisco CA, USA, 2013, pp. 2333–2338. doi: 10.1145/2505515.2505665.

[20] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A factorization-machine based neural network for CTR prediction," in *Int. Joint Conf. Artif. Intell.*, Melbourne, Australia, 2017, pp. 1725–1731. doi: 10.48550/arXiv.1703.04247.

[21] X. He and T. S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, Shinjuku Tokyo, Japan, 2017, pp. 355–364. doi: 10.1145/3077136.3080777.

[22] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. Int. Conf. Mach. Learn.*, Beijing, China, 2014, pp. 1188–1196. doi: 10.48550/arXiv.1405.4053.

[23] F. Hill, K. Cho, and A. Korhonen, "Learning distributed representations of sentences from unlabeled data," in *Proc. 2016 Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, San Diego, CA, USA, 2016, pp. 12–17. doi: 10.48550/arXiv.1602.03483.

[24] S. Arora, Y. Liang, and T. Ma, "A simple but tough-to-beat baseline for sentence embeddings," in *Int. Conf. Learn. Representations*, Puerto Rico, USA, 2016.

[25] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Int. Conf. Learn. Representations*, San Diego, CA, USA, 2015. doi: 10.48550/arXiv.1409.0473.

[26] A. Vaswani, N. Shazeer, N. Parmar, and J. Uszkoreit, "Attention is all you need," in *Neural Inf. Process. Syst.*, Long Beach, CA, USA, 2017, pp. 5998–6008.

[27] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018. doi: 10.48550/arXiv.1810.04805.

[28] N. Y. Wang, Y. X. Ye, L. Liu, L. Z. Feng, T. Bao and T. Peng, “Language models based on deep learning: A review,” *J. Softw.*, vol. 32, no. 4, pp. 1082–1115, 2020.

[29] S. Kim, N. Lee, D. Kim, M. Yang, and C. Park, “Task relation-aware continual user representation learning,” in *Proc. 29th ACM SIGKDD Conference on Knowl. Discovery Data Mining*, Long Beach, CA, USA, 2023, pp. pp 1107–1119. doi: 10.1145/3580305.3599516.

[30] L. Xue, D. Yang, S. Zhai, Y. Li, and Y. Xiao, “Learning dual-view user representations for enhanced sequential recommendation,” *ACM Trans. Inf. Syst.*, vol. 41, no. 4, pp 1–26, 2022.

[31] B. Wu, Y. Kang, B. Guan, and Y. Wang, “We are not so similar: Alleviating user representation collapse in social recommendation,” in *Proc. 2023 ACM Int. Conf. Multimedia Retrieval*, Thessaloniki, Greece, 2023, pp. 378–387. doi: 10.1145/3591106.3592244.

[32] C. Li *et al.*, “RecGURU: Adversarial learning of generalized user representations for cross-domain recommendation,” in *WSDM ‘22: Proc. Tenth ACM Int. Conf. Web Search and Data Mining*, 2022, pp. 571–581. doi: 10.1145/3488560.3498388.

[33] L. Gong, L. Lin, W. Song, and H. Wang, “JNET: Learning user representations via joint network embedding and topic embedding,” in *WSDM ’20: Proc. Tenth ACM Int. Conf. Web Search and Data Mining*, Houston TX, USA, 2020, pp. 205–213. doi: 10.1145/3336191.3371770.

[34] H. Wang, P. Li, W. Tao, B. Feng, and J. Shao, “Learning dynamic user behavior based on error-driven event representation,” in *WWW ’21: Proc. Web Conf. 2021*, Ljubljana, Slovenia, 2021, pp. 2457–2465. doi: 10.1145/3442381.3450012.

[35] M. Pagliardini, P. Gupta, and M. Jaggi, “Unsupervised learning of sentence embeddings using compositional n-gram features,” arXiv preprint arXiv:1703.02507, 2017. doi: 10.48550/arXiv.1703.02507.

[36] Y. J. Zhang, Z. Dong, and X. W. Meng, “Research on personalized advertising recommendation systems and their applications,” (in Chinese), *Chin. J. Comput.*, vol. 44, no. 3, pp. 531–563, 2021. doi: 10.11897/SP.J.1016.2021.00531.

[37] J. W. Ahn, P. Brusilovsky, J. Grady, D. He, and S. Y. Syn, “Open user profiles for adaptive news systems: Help or harm?,” in *Proc. 16th Int. Conf. World Wide Web*, Banff Alberta, Canada, 2007, pp. 11–20. doi: 10.1145/1242572.1242575.

[38] L. Zheng, V. Noroozi, and P. S. Yu, “Joint deep modeling of users and items using reviews for recommendation,” in *WSDM ’17: Proc. Tenth ACM Int. Conf. Web Search and Data Mining*, Cambridge, UK, 2017, pp. 425–434. doi: 10.1145/3018661.3018665.

[39] J. Xie, F. X. Zhu, X. F. Li, S. Huang, and S. C. Liu, “Attentive preference personalized recommendation with sentence-level explanations,” *Neurocomputing*, vol. 426, no. 2, pp. 235–247, 2021. doi: 10.1016/j.neucom.2020.10.041.

[40] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, “Session-based recommendations with recurrent neural networks,” in *Int. Conf. Learn. Representations*, 2015. doi: 10.48550/arXiv.1511.06939.

[41] S. Zhang, Y. Tay, L. Yao, A. Sun, and J. An, “Next item recommendation with self-attentive metric learning,” in *Thirty-Third AAAI Conf. Artif. Intell.*, Hawaii, USA, 2019.

[42] J. Gao *et al.*, “Set-sequence-graph: A multi-view approach towards exploiting reviews for recommendation,” in *CIKM ’20: Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Ireland, 2020, pp. 395–404. doi: 10.1145/3340531.3411939.