

Received September 28, 2020, accepted October 12, 2020, date of publication October 15, 2020, date of current version October 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3031281

# A Hybrid Model Based on LFM and BiGRU Toward Research Paper Recommendation

XU ZHAO<sup>1</sup>, HUI KANG<sup>1,2,3</sup>, TIE FENG<sup>1,2</sup>, CHENKUN MENG<sup>1</sup>, AND ZIQING NIE<sup>2</sup>

<sup>1</sup>College of Software, Jilin University, Changchun 130012, China

<sup>2</sup>College of Computer Science and Technology, Jilin University, Changchun 130012, China

<sup>3</sup>Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, Jilin University, Changchun 130012, China

Corresponding author: Tie Feng (fengtief@jlu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC0831706, in part by the Science and Technology Development Plan Project of Jilin Province under Grant 20200201166JC, and in part by the Innovation and Entrepreneurship Training Program of Jilin University under Grant 202010183495.

**ABSTRACT** To improve the accuracy of user implicit rating prediction, we combine the traditional latent factor model (LFM) and bidirectional gated recurrent unit neural network (BiGRU) model to propose a hybrid model that deeply mines the latent semantics in the unstructured content of the text and generates a more accurate rating matrix. First, we utilize the user's historical behavior (favorites records) to build a user rating matrix and decompose the matrix to obtain the latent factor vectors of users and literature. We also apply the BERT model for word embedding of the research papers to obtain the sequence of word vectors. Then, we apply the BiGRU with the user attention mechanism to mine the research paper textual content and to generate the new literature latent feature vectors that are used to replace the original literature latent factor vectors decomposed from the rating matrix. Finally, a new rating matrix is generated to obtain users' ratings of noninteractive research papers and to generate the recommendation list according to the user latent factor vector. We design experiments on the real datasets and verify that the research paper recommendation model is superior to traditional recommendation models in terms of precision, recall, F1-value, coverage, popularity and diversity.

**INDEX TERMS** Recommender systems, deep learning, LFM, BiGRU, user attention.


## I. INTRODUCTION

The Internet currently provides us with abundant online content, which makes it very time consuming to go over every detail and find needed information. This is often referred to as the information overload problem [1], where users find an overwhelming number of publications that match their search queries but are largely irrelevant to their latent information needs [2]. Researchers have encountered the problem of information overload while consulting research paper data. For researchers, how to quickly find papers of interest is a considerable challenge. To solve this problem, search engines and recommender systems are widely investigated.

Search engines return relevant content based on keywords provided by users. Recommender systems (RSs) are rapidly becoming key instruments in solving the problem of information overload [3], and its core is the recommendation

algorithm. The main task of the recommendation algorithm is to connect users and information in a certain way to help users find the information that they may be interested in [4]. Learning from the success of RSs in many different domains, such as movies, news, and social networks, relevant recommendation techniques can be applied to the domain of research paper recommendation. By analyzing the historical behavior data (favorites records) of researchers, we can model their research preferences and actively recommend research papers they are interested to solve the problem of information overload in academic research and to reduce the research cost.

In the past few decades, deep learning technology has made breakthrough progress in the fields of computer vision, natural language processing, speech recognition, machine translation, and online advertising [5]. Deep models tend to perform well in situations where the datasets are well characterized and can be trained on a large quantity of appropriately labeled data [6]. Deep learning-based RS usually receives user- and item-related data and uses deep neural networks [7] to extract

The associate editor coordinating the review of this manuscript and approving it for publication was Farhana Jabeen Jabeen .

user and item features; then, it combines with traditional recommendation algorithms to generate item recommendations. Existing deep learning-based research paper RSs mostly use deep learning models to mine researchers' historical behavior information (e.g., citations, downloads, and favorites records) to model user interests. However, most of the current research paper RSs are not sufficient for mining the textual content of the research papers. The use of one-hot codes for word embedding leads to an inability to reflect the difference in words at the part-of-speech and semantic level. The vector space model (such as the bag-of-words model) does not deeply mine the serialized features of words (such as long-distance phrase structure) when encoding text sequences. These inadequacies cause the extracted paper feature representation to be inaccurate, which affects the recommendation performance of the RS for research papers.

In the information retrieval scenario, the correctness and ranking of search results greatly affect the user's search experience [8]. Users often want the correct recommendation results to appear first in the entire list. In the recommendation scenario, for the reason that the user's needs are not clearly given, the RS needs to infer the user's interests (implicit, diverse) and find the user's favorite items. The recommended literature must not only meet the user's interests but also have a certain degree of difference and diversity to meet the diverse needs of the user.

Aiming at the problems of insufficient text mining and diversity of recommendation results, we combine the traditional LFM and BiGRU model to propose a hybrid model for research paper recommendation. It is divided into three modules. In the first module, the traditional LFM is used to extract latent factor vectors of users and literature from the user-to-literature interaction matrix [9]. In the second module, word vector technology is applied to store rich text semantic information and word order position information. It helps to enhance text data representation of literature and to solve the problem of lack of effective data representation in the traditional hybrid recommendation. We adopt the BiGRU model with a user attention mechanism to learn the latent feature vectors that contain content information from the research paper textual content and are used instead of the latent factor vector learned by the LFM to improve the accuracy of the implicit rating prediction. In the third module, we set weights for user ratings and document diversity to generate a recommendation list for research papers.

The main contributions of our proposed scheme are summarized as follows:

- 1) We combine traditional LFM and BiGRU to propose a hybrid neural network model for research paper recommendation.
- 2) We apply a pooling approach based on a user attention mechanism to assign attention weights to each word in the sequence of word vectors to extract latent literature features that can reflect both user preferences and the content of the research paper.

- 3) In the construction of the literature latent factor vector, we assign basic weights to keywords in literature in advance to prevent the subject information of literature in word vectors from being deleted after being pooled.
- 4) In the recommendation list generation stage, we utilize a rating matrix to generate the recommendation list according to the user latent factor vector and recommend multiple types of documents to meet the different needs of users under the premise of ensuring precision.

The remainder of the paper is organized as follows. Section II reviews the related work. Section III introduces a hybrid recommendation model based on deep learning for the research paper. Section IV compares the experimental results, and Section V presents a summary of the findings and conclusions.

## II. RELATED WORK

Since the GroupLens research group of the University of Minnesota first proposed the collaborative filtering recommendation algorithm in the 1990s, RS has become a research hotspot in the computer field. In the past two decades, many scholars and enterprises have studied RS in depth and proposed many new methods and technologies [4], [10], [11]. These research results have been widely utilized in various fields. In the field of research paper recommendation, researchers worldwide are currently focusing on the structure and semantic information of research papers.

### A. RESEARCH PAPER RECOMMENDATION BASED ON COLLABORATIVE FILTERING

The technique of collaborative filtering (CF) is especially successful in generating personalized recommendations [12]. The goal of CF is to predict the preferences of one user, referred to as the active user, based on the preferences of a group of users. To integrate the collaborative filtering algorithm into the field of research paper recommendation, McNeer *et al.* [13] analogized between a research paper RS and a movie RS. They analyzed the citations of researchers on research paper to construct a researcher-literature rating matrix, and then they processed the rating matrix through principal component analysis (PCA) and other dimensionality reduction techniques to achieve a recommendation for research papers. However, that would be less suitable for finding closely related references but perhaps more suitable for finding novel references for users. Pennock *et al.* [14] suggested a method called personality diagnosis (PD) to recommend research papers to similar users. This method calculates the similarity between users based on the user's preference for the item, then it calculates the probability that the users like the new items and recommends research papers with a higher probability value to the user. This method retains the advantages of the traditional similarity calculation and additionally supports the addition of new data to achieve good practical results. One of the limitations with this method is that it excessively relies on how users rate titles.

## B. CONTENT-BASED RESEARCH PAPER RECOMMENDATION

The difficulty of the content-based research paper recommendation algorithm is how to find the content feature representation [15] that can reflect the professional semantics of the research papers. Sugiyama and Kan [16] generated academic paper recommendation results based on users' recent research interests, which improved the recommendation effect. Kazemi and Abhari [17] compared the efficiency and usability differences between two well-known content-based recommendation methods of term frequency inverse document frequency (TF-IDF) in the abstract extraction of the paper and the feature generation of the recommendation system extraction. The most serious disadvantage of this method is that it only focuses on the abstracts of research papers.

## C. RECOMMENDATION SYSTEM BASED ON HYBRID RECOMMENDATION

The hybrid recommendation algorithm usually integrates different recommendation models to complement each other to achieve better recommendation results [18]. Basu *et al.* [19] proposed a method that can simultaneously apply citation information and content information to calculate the similarity between two papers. They applied keyword vectors to construct paper features to achieve research paper recommendations. However, they did not consider the impact of multiple information sources, including sources that exploit a limited amount of content. The performance of traditional recommendation algorithms when mining and analyzing item content and user historical interaction records to model user interests and item features are still limited [20]. Finding a more effective feature extractor is crucial to improving the precision of the recommendation system [21]. McAuley *et al.* combined ratings with review text to propose a hidden factors and hidden topics model (HFT) for product recommendations. Their approach can fit user and product parameters with only a few reviews [22]. Wang and Blei [23] combined the merits of traditional collaborative filtering and probabilistic topic model to propose a collaborative topic regression (CTR) model for scientific articles recommendations. This method has improved the recall rate of literature recommendation. Nevertheless, the latent representation learned by CTR may not be very effective when the auxiliary information is very sparse. Kim *et al.* [24] proposed a context-aware recommendation model which integrates Convolutional Neural Network (CNN) into LMF in order to capture contextual information in description documents for the rating prediction. Zhang *et al.* [25] proposed an extreme residual connected convolution collaborative filtering (xRConvCF) model which utilizes textual information to predict rating for each item. This model mitigates the problem of the vanishing gradient and enhance feature reuse. Cheng *et al.* [26] proposed an aspect-aware latent factor model (ALFM), which could effectively combine reviews and ratings for rating

prediction. Their model could alleviate the data sparsity problem and gain good interpretability for recommendation. Based on their own previous research, they proposed a multi-modal aspect-aware latent factor model (MMALFM) for rating prediction and investigating the utility of item images on the performance [27].

## D. RECOMMENDATION SYSTEM BASED ON DEEP LEARNING

Compared with the traditional recommendation model, deep learning technology [28] can mine research paper features and user interaction records more deeply. On the one hand, through self-representation learning, the deep features of user and project data can be learned and represented [29]; on the other hand, through automatic feature learning from multi-source heterogeneous data, there is no need to map the data to the same latent space. Sedhain *et al.* [30] combined the deep self-encoder model and shallow collaborative filtering recommendation method, and they proposed the AutoRec recommendation model. It takes the user's rating matrix as input and generates a reconstructed rating matrix after passing through an encoder and a decoder and trains the model by minimizing the error of the parameters. The intermediate results of the model can be regarded as latent for user and item vectors. The collaborative deep learning (CDL) model proposed by Wang [31] combined the Bayesian stacked denoising auto encoder (SDAE) and the probabilistic matrix decomposition (PMF) in a tightly coupled manner to capture the similarities and deep layers between item content and user relationships. However, overall, since the training is disjoint, each individual model size usually needs to be larger to achieve reasonable accuracy for an ensemble to work. Combined with attention neural network, they presented an aspect-aware recommender model named A3NCF for rating prediction [32]. Chin *et al.* proposed an end-to-end Aspect-based Neural Recommender (ANR) to perform aspect-based representation learning for both users and items via an attention-based component [33].

Although deep neural networks can effectively capture the nonlinear relationship between users and items, the premise of applying deep neural networks is that each user or item is subject to independent and identical distributions, which is obviously contradictory to reality. Therefore, the deep learning model is slightly inferior to the collaborative filtering shallow model in capturing user interest and the matching degree and potential relationship between items. To extract a more accurate representation of the literature features and capture the potential relationship between user interests and research papers, we combine the content-based recommendation method with the LFM. The BiGRU based on the user attention mechanism can further help the network remember more important information for users and can provide better interpretability for feature extraction of textual content information, improving the precision of recommendation results and user satisfaction.

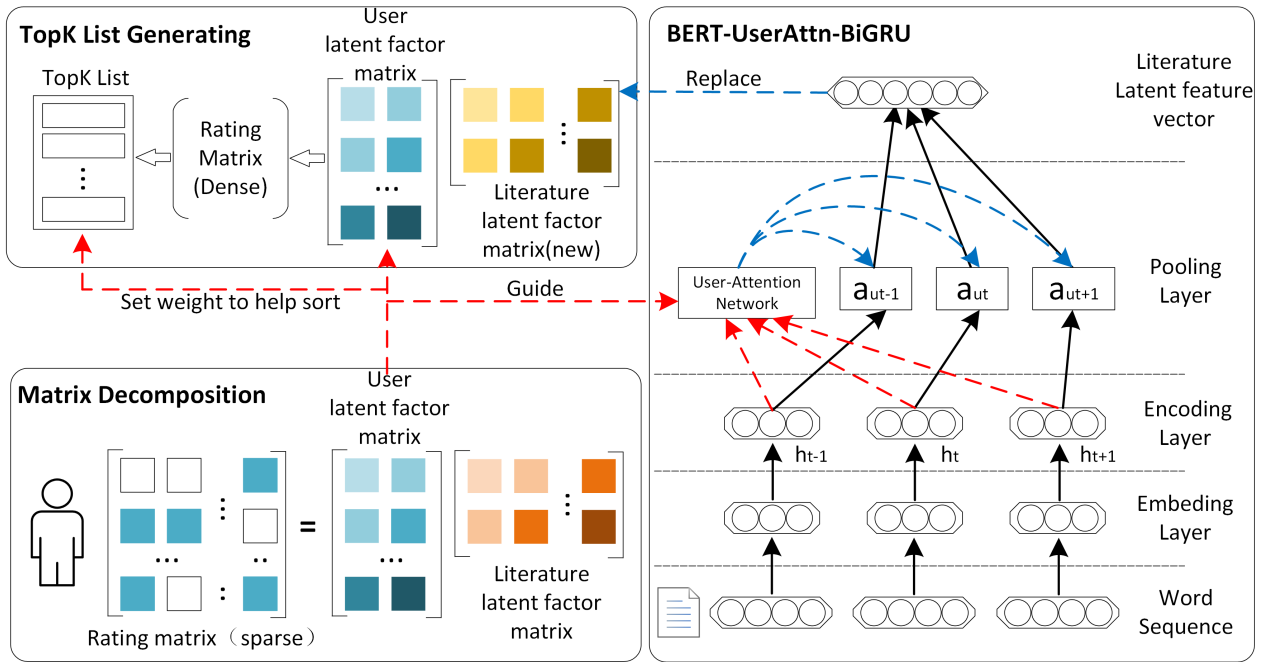


FIGURE 1. The hybrid recommendation model for research papers.

### III. HYBRID RECOMMENDATION MODEL BASED ON DEEP LEARNING

In this section, we first formulate a hybrid recommendation model based on deep learning for further decentralized research and then introduce the details of our recommendation model.

For the purpose of offsetting the excessive specialization of the recommendation results caused by the content-based recommendation method and the problems of sparse data and cold start generated by the collaborative filtering-based recommendation method, we unify the two methods for hybrid recommendations. The model structure is shown in Fig. 1. First, we utilize the LFM to extract the user latent factors and literature latent factors that reflect the user’s interests and then use the BiGRU with the attention mechanism to deeply mine the research paper textual content. However, there is a slight deviation between the feature representation extracted by BiGRU reflecting the content information of the literature and the literature latent factors. For the purpose of minimizing this deviation, we employ the literature latent factor vectors to guide the training process of the deep neural networks and use the user latent factors to guide the dynamic weight distribution of the word vector in the process of extracting the literature features. Finally, we directly match the user latent factors with the high-level feature representation of the extracted paper by the deep neural networks to predict the user’s preference for candidate research papers. The collaborative filtering method learns from the user’s historical behavior (favorites record), guides and adjusts the content feature extraction process of the content-based recommendation method. The extracted paper content features can also modify the literature

latent factors generated by the LFM method to improve the recommendation performance of the collaborative filtering method. When generating a research paper recommendation list, we sort the user’s preference literature from high to low and recommend them in order. The model mainly uses BERT word embedding technology, sequence coding technology based on the BiGRU network and pooling technology based on the user attention mechanism.

#### A. MATRIX DECOMPOSITION BASED ON LFM

The latent factor model [34] is a generalization of content-based filtering. LFM first classifies all items and then recommends items to users based on the user’s classification of interest [35]. The matrix decomposition in Fig. 1 depicts the standard LFM. Now,  $r_{u,i}$  forms a ratings matrix by taking users as rows and items as columns. The entire ratings matrix is expressed as the product of the user latent factor matrix and the literature latent factor matrix. As a product of two matrices, it is expressed mathematically in the form of matrix decomposition. The rating of user  $i$  on item  $j$  is modeled as:

$$r_{u,i} = p_u q_i = \sum_{k=1}^F p_{u,k} q_{i,k}^T \quad (1)$$

where  $p_{u,k}$  denotes the relationship between the interest of user  $u$  and the  $k$ -th latent factor, and  $q_{i,k}$  denotes the relationship between item  $i$  and the  $k$ -th latent factor.  $F$  denotes the number of latent factors, and  $r$  is the user’s interest in item [36].

As shown in Algorithm 1, in the latent factor matrix decomposition stage, the user latent factor matrix and

**Algorithm 1** LFM

---

```

1 Input: the rating matrix  $ratings$ , the number of iterations
   $N$ , the number of latent factors  $K$ , the regular term  $\lambda$  and
  the stride  $\alpha$ ;
2 Output: the user latent factor matrix  $P_{u,k}$ , literature
  latent factor matrix  $Q_{i,k}$ ;
3 Initialize the matrices  $P_{user}$  and  $Q_{item}$  randomly;
4 Stochastic gradient descent method training parameters
   $P_{user}$  and  $Q_{item}$ ;
5 for  $step \leftarrow 1$  in  $range(0, N)$  do
6   for  $user, item$  in  $ratings$  do
7     Randomly draw negative  $samples$ ;
8      $I_{u,i} \leftarrow 0$ ;
9     for  $item, r_{u,i}$  in  $items$  do
10       $\hat{r}_{u,i} \leftarrow Predict\ the\ user's\ rating$ ;
11       $err_{u,i} \leftarrow r_{u,i} - \hat{r}_{u,i}$ ;
12       $I_{u,i} \leftarrow I_{u,i} + \alpha * err_{u,i}$ ;
13      for  $f \leftarrow 0$  in  $range(0, F)$  do
14         $P_{u,k} \leftarrow$ 
15          $P_{u,k} + \alpha * (I_{u,i} * err_{u,i} * Q_{i,k} - \lambda * P_{u,k})$ ;
16          $Q_{i,k} \leftarrow$ 
17          $Q_{i,k} + \alpha * (I_{u,i} * err_{u,i} * P_{u,k} - \lambda * Q_{i,k})$ ;
18      end
19    end
20  end
   $\alpha \leftarrow \alpha * 0.9$ ;

```

---

research paper latent factor matrix are randomly initialized on the divided training set, and the inner product of the two is used to represent the user's predicted rating for the candidate research papers. The goal is to minimize the mean square error between the predicted rating and the actual rating in the training set. In the dataset, since the user's interaction records for items are often concentrated in a certain class or several classes, the difference in the scores of similar research papers in the same class is stable. Therefore, we set the bias term  $I_{u,i}$  based on the original loss function to decrease the number of iterations. The loss function  $f_{loss}$  is expressed as:

$$f_{loss} = \frac{1}{2} \sum_{(u,i) \in K} I_{u,i} (r_{u,i} - \sum_{k=1}^k p_{u,k} q_{i,k}^T)^2 + \frac{\lambda}{2} \sum_{k=1}^k \|p_{u,k}\|^2 + \frac{\lambda}{2} \sum_{k=1}^k \|q_{i,k}\|^2 \quad (2)$$

where bias term  $I_{u,i} = \frac{b_{u,i}}{\mu}$ ,  $b_{u,i}$  indicates the number of user interactions with the item, and  $\mu$  denotes the global average of all rating records of user  $i$ .  $\lambda \|p_{u,k}\|^2$  and  $\lambda \|q_{i,k}\|^2$  in the above formula denote the regularization terms used to prevent overfitting.  $\lambda$  shows the regularization parameter, which needs to be obtained through repeated experiments according to specific application scenarios. The optimization of the loss

function uses a stochastic gradient descent algorithm:

$$p_{u,k}^{f+1} = p_{u,k}^f + \alpha (I_{u,i} (r_{u,i} - \sum_{k=1}^k p_{u,k} q_{i,k}^T) q_{i,k}^T - \lambda p_{u,k}^f) \quad (3)$$

$$q_{i,k}^{f+1} = q_{i,k}^f + \alpha (I_{u,i} (r_{u,i} - \sum_{k=1}^k p_{u,k} q_{i,k}^T) p_{u,k} - \lambda q_{i,k}^f) \quad (4)$$

Iterative calculation continuously optimizes the parameters (manually sets the number of iterations in advance) until the parameters converge. Finally, we obtain the user latent factor matrix  $P_{u,k}$  and literature latent factor matrix  $Q_{i,k}$ , which can reflect the user's interest preference.

**B. BERT WORD EMBEDDING BASED ON PRETRAINING**

Before using the text content as the input of the model, we preprocess the text content data according to the following steps: (1) In order to avoid the model from recognizing the same word as different words due to the difference in capitalization, the words in the text are unified expressed in lowercase letters. (2) We identify the separators such as spaces and punctuation marks in the text content, and then we use these characters to divide the text content into independent words. (3) The stop words are removed. (4) The words received are summarized into a dictionary. (5) According to the generated dictionary, the index of each word in the dictionary is obtained, and a one-hot code for each word is generated. (6) The word vector sequence of the literature are obtained.

The BERT model pre-trained takes preprocessed literature content as input. First, we add the word embedding, syntactic embedding, and position embedding of the word sequence through the embedding layer and then solve the masking language model task through the two-way transformer model. Finally, we extract the word vector sequences from the titles and abstracts of the research papers.

**1) INPUT LAYER**

The research paper textual content can be regarded as a sequence of words, denoted by  $X = (x_1, x_2, \dots, x_T)$ , where  $T$  indicates the length of the text sequence, and the word sequence corresponding to the textual content of different literature is also different.  $x_t \in R^V$  represents the word  $t$  in the text sequence and is usually expressed in the form of a one-hot code, where  $V$  indicates the size of the dictionary of the dataset. Therefore,  $x_t$  denotes a  $V$ -dimensional vector, the  $t$  th element of this vector is 1, and all other elements are 0. It should be noted that the representation of the text sequence  $X$  here strictly preserves the order of words in the textual content, which provides timing information for subsequent semantic coding based on the BiGRU network.

**2) WORD EMBEDDING LAYER**

The second layer of the network is the word embedding layer. Since the scale of the dataset dictionary may reach 100,000 or even one million levels, using a single hot code to

represent words may cause a problem of dimensional disaster, which is not conducive to the processing of subsequent tasks. Therefore, we apply the word embedding layer to convert the high-dimensional one-hot vector  $x_t$  of the word into a low-dimensional word vector  $e_t$  through the pretrained BERT network.

### C. WORD SEQUENCE CODING BASED ON BiGRU

The standard GRU can only capture the forward semantic information of the text sequence, and in practical applications, the reverse text sequence implies the available semantic information that has mining potential [37]. Therefore, we utilize a BiGRU to simultaneously model the forward and reverse sequences of the textual content of the literature to obtain a more comprehensive and accurate text sequence encoding. BiGRU transforms the GRU network structure, changing the original one-way feedforward neural network to a two-way neural network, encoding the context of the literature from the forward and reverse directions, respectively, and according to different weights at each time step. The hidden states of the two networks are added together as the hidden state of the bidirectional network at the moment to obtain a more effective representation of the literature features. To save the state of these two hidden layers simultaneously, BiGRU needs twice the storage space to save their weights and offset parameters.

When training the GRU networks, we take the word vector sequences output by the BERT model as input and set the hidden layer dimension for GRU. Since GRU implements its own iteration, there is no need to specify the number of time steps. We apply the literature latent factor matrix learned in the LFM as the label of the training set, with the aim of minimizing the mean square deviation of the literature latent factor matrix and the hidden state sequence. Then, we apply the Adam algorithm [38] for small batch gradient descent during training, which continuously updates the model parameters until the algorithm converges.

### D. POOLING TECHNOLOGY BASED ON THE USER ATTENTION MECHANISM

In recent years, attention-based neural network architectures, which learn to focus their “attention” on specific parts of the input, have shown promising results on various tasks [39]–[41]. The pooling method based on the user’s attention mechanism fully considers the weight of the extracted text feature vectors of the user’s preferences, unifies the content-based recommendation method and the collaborative filtering method, and improves the accuracy of the recommendation.

In this work, due to the word sequence of the text is long, the final feature vector obtained by the BiGRU model in actual applications will be biased toward the last few words of the text sequence. To solve this problem, we adopt an attention mechanism, which can make full use of user preferences to adjust the weight of word vectors to obtain a more compact and accurate representation of content features. The main idea of the user attention mechanism in this work is

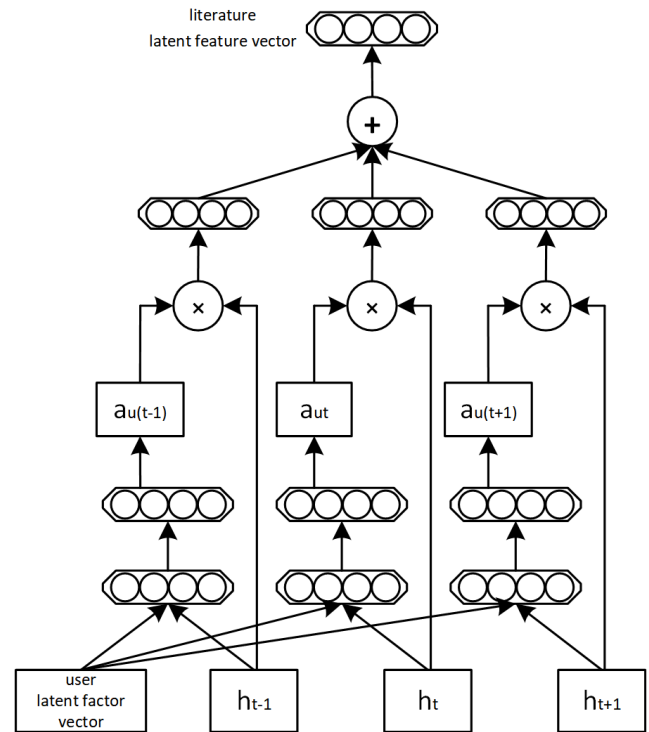


FIGURE 2. The network structure of the pooling layer based on the user attention mechanism.

to automatically examine the importance of different words in the text sequence to different users through the attention network when generating text feature vectors according to the degree of user latent factors affecting the words in the text sequence and assign different weights to the words. The weighted average of the word vector sequences is used to represent the text feature vector of the literature. The obtained text feature vector can not only retain the textual content of the literature but also reflect the user’s preferences, which is more suitable for the next recommendation task. The network structure of the pooling layer based on the user attention mechanism is shown in Fig. 2.

The user attention network takes the user latent factors  $p_u$  learned by the shallow LFM and the word sequences  $h_t$  obtained by the word encoder based on the BiGRU network as input. First, at time  $t$ , the weight matrices  $W^{(u)}$  and  $W^{(h)}$  are used to linearly transform  $p_u$  and  $h_t$  in the hidden state, respectively. Then, the nonlinear activation function is applied to extract the nonlinear semantic information. Finally, linear transformation is used again to obtain the attention degree of user  $u$  to word  $t$  in the text sequence. The formula for the calculation process is as follows:

$$r_{u,t} = R(p_u, h_t) = v^T \tanh(W^{(u)}p_u + W^{(h)}h_t) \quad (5)$$

where the vector  $v$ , weight matrix  $W^{(u)}$  and  $W^{(h)}$  can be learned by training the user attention network.

According to the attention degree of user  $u$  to word  $t$  in the text sequence, the normalization operation is performed

to obtain weight  $a_{ut}$  assigned by user  $u$  to word  $t$  in the text. According to the weight value output by the user attention network, the weighted average  $q_u = \sum_{t=1}^T a_{ut}h_t$  of the word vector sequence is used to represent the new literature latent factors extracted from the content of the literature. The new literature latent factors are applied to replace the original literature latent factors in the next stage.

### E. RECOMMENDATION LIST GENERATION

After obtaining the latent factor vector matrix of users and literature, the user rating matrix can be generated by LFM. The RS allows a certain type of content to appear together, which is not effective for users with diverse needs. When the number of recommended documents is small, users tend to focus on the documents in a certain category of interest; when the number of recommendations increases, the diversity of recommended documents gradually becomes important for many types of interested users. Additionally, in the recommendation results, a certain article scores low in the class that users care about, but it scores high in the class with low user interest. This article will not attract users under normal circumstances, but it will eventually appear in the final recommendation list due to the high overall score.

To solve the above two problems:

- 1) To meet the diversity of recommended results, in the scenario of recommending multiple research papers, under the premise of ensuring the accuracy of the recommended results, the RS should limit the large number of research papers of the same type from appearing in the results.
- 2) To make the correct recommendation result rank high, we rescored the literature by setting weights and generating the recommendation results.

As shown in Algorithm 2, the reordering process is as follows. First, original user ratings were removed from the new dense ratings matrix to prevent recommending literature that the user has already viewed. For a certain user, we sort the scores in descending order and intercept the first  $n$  as the candidate set. Then, we calculate the new user-item score, and the formula is as follows:

$$r'_{u,i} = (1 - W^{(K)})r_{u,i} + W^{(K)}d_{k,i} \quad (6)$$

where  $r'_{u,i}$  denotes the user's new rating for the literature,  $W^{(K)} = \frac{p_{u,k}}{\sum p_{u,k}}$  indicates the weight of a certain class that the user is interested in, and  $d_{k,i}$  denotes the diversity of literature  $i$ .

Finally, according to the new score sorting, a recommendation list is generated that focuses on mining the original recommendation results, so if the accuracy of the original results is high, it will not have a greater impact on the recommendation results.

## IV. EXPERIMENTS

In this section, we establish several comparative experiments and control experiments respectively to compare and evaluate

### Algorithm 2 Recommended List Generating

---

```

1 Input: the original rating matrix  $r_{u,i}$ ;
2 Output: the new rating matrix  $r'_{u,i}$ ;
3 Define and initialize the related variables: the weight of
  a certain class  $W^{(K)}$  and the diversity of literature  $d_{k,i}$ ;
4 Remove original user ratings from the new ratings
  matrix and sort the scores in desc order;
5 for  $r_{u,i}$  in the candidate set do
6    $W^{(K)} \leftarrow 0$ ;
7   for  $p_t$  in  $p_{u,k}$  do
8     if  $p_t$  is the maximum then
9        $W^{(K)} \leftarrow p_t / \text{sum}(p_{u,k})$ ;
10    end
11  end
12  for  $q_s$  in  $q_{i,k}$  do
13    if  $q_s$  is the maximum then
14       $d_{k,i} \leftarrow 1 - q_s / \text{sum}(q_{i,k})$ ;
15    end
16  end
17   $r'_{u,i} \leftarrow (1 - W^{(K)})r_{u,i} + W^{(K)}d_{k,i}$ ;
18 end

```

---

the proposed model on six metrics including precision. First, the datasets applied in the experiments is introduced. Then, several methods and models for comparison are presented. Finally, the experimental results obtained are analyzed and discussed.

In the experiment, the Keras deep learning framework is used to implement the hybrid research paper recommendation model. In the latent factor matrix decomposition stage, the user latent factor matrix and research paper latent factor matrix are randomly initialized on the divided training set. We employ the validation set to determine the optimal hyperparameters that  $\lambda = 0.01$ ,  $\alpha = 0.01$  and  $F = 150$ . For BERT model, the pre-training procedure largely follows the existing literature on language model pre-training. Devlin et al. [42] used the BooksCorpus (800M words) [43] and English Wikipedia (2,500M words) for the pre-training corpus. In our experiment, we utilize uncased\_L12\_H768\_A-12<sup>1</sup> pre-trained model of Google to map the words in the literature to 768-dimensional word vectors. For training BiGRU networks, we use the literature latent factor learned in the LFM as the label of the training set. Aiming to minimize the standard deviation between the literature latent factor and the hidden state sequence, the Adam algorithm [44] is applied for small batch gradient descent training, and the model parameters are continuously updated until the algorithm converges. For BiGRU network, the num of layers is set to 3, the learning rate is set to 0.01 and the number of epochs is 30. Besides, the dropout technique [45] is used to prevent overfitting and the dropout ratio is 0.5.

<sup>1</sup><https://github.com/google-research/bert>

## A. DATASETS

### 1) CiteULike-a

<sup>2</sup> This dataset consists of two tables, user info and raw data. The user-info table includes 204,986 favorite records of 16,980 articles by approximately 5,551 users, which are expressed in the form of a predicted value of 1. The raw data table includes the number, title, and abstract of 16,980 articles on the CiteULike website.

### 2) LibraryThing

<sup>3</sup> There are 120,150 books (including book titles and abstracts), user information and ratings in the LibraryThing dataset. After deleting users with less than 10 ratings, we got 185,210 favorites records, 150,216 ratings, and 139,530 reviews of 12,350 users.

By analyzing the CiteULike-a dataset, it can be found that it seriously lacks negative samples. It is necessary to randomly collect negative samples to expand the experimental dataset. The process is as follows. First, we select negative samples from the articles that the user has not collected. Then, we randomly select articles with the same number of articles collected by the user from the candidate pool, and we assign the user's prediction value to them as 0. Finally, we insert the negative samples into the original user-info table according to the user ID, and we employ the shuffle method to shuffle the order of records to obtain a new user behavior data table.

The process of dividing the dataset is described as follows. First, we randomly extract 20% from each user's behavior data as the test set and the remaining 80% as the training set. Second, we utilize the 50% cross-validation method to randomly divide the experimental dataset into 5 subsets. One of the 5 subsets is used as the validation set, and the other 4 subsets are applied as the training sets to repeat the experiment 5 times. In the experiment, the recommended model is trained on the divided training sets, after which the performance of the recommended model is evaluated on the dataset.

## B. EVALUATION METRICS

In the literature recommendation task, the precision rate represents the proportion of the number of samples that are correctly predicted to the total number of samples. For each user, the definition of prediction precision is shown as follows:

$$Precision = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|} \quad (7)$$

where  $R(u)$  indicates the recommendation set generated for the user  $u$ , and  $T(u)$  indicates the literature in user's favorites.

The recall rate indicates the proportion of number of samples that are correctly predicted to the total number of samples of literature in user's library. For each user, the definition of

prediction precision is shown as follows:

$$Recall = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|} \quad (8)$$

$F1$  represents the harmonic mean of precision and recall, the definition of prediction precision is shown as follows:

$$F1 = \frac{2 * Recall * Precision}{Recall + Precision} \quad (9)$$

The coverage rate is used to evaluate the ability of the recommendation system to discover the long tail of items. The higher the coverage rate, the better the ability of the recommendation algorithm to discover the long tail of items and the formula is as follows:

$$Coverage = \frac{|\cup_{u \in U} R(u)|}{I} \quad (10)$$

where  $U$  denotes the user set and  $I$  indicates total item set.

The Popularity of an item refers to how many users rate it. We apply the average popularity of items to evaluate the novelty of the recommended results. Since the popularity distribution of items satisfies the long-tailed distribution, we take the logarithm of the popularity of each item when calculating the average popularity.

Finally, we apply the intra-list similarity (ILS) to evaluate Algorithm 2 proposed in Section III. The ILS describes the dissimilarity between items in the recommended list, and the formula is as follows:

$$ILS = \frac{\sum_{b_i \in N} \sum_{b_j \in N, b_i \neq b_j} S(b_i, b_j)}{\sum_{b_i \in N} \sum_{b_j \in N, b_i \neq b_j} 1} \quad (11)$$

where  $S(b_i, b_j)$  indicates the cosine similarity of two items  $b_i$  and  $b_j$ . The smaller the ILS, the more diverse the literature in the recommendation results.

## C. BASELINE METHODS

According to the techniques we applied, the hybrid recommendation model proposed in this paper is referred to as the BAGM (BERT-userAttn-BiGRU-LFM). We evaluate the BAGM and baseline methods in terms of precision, recall, F1-value, coverage, popularity and diversity.

### 1) VARIANTS

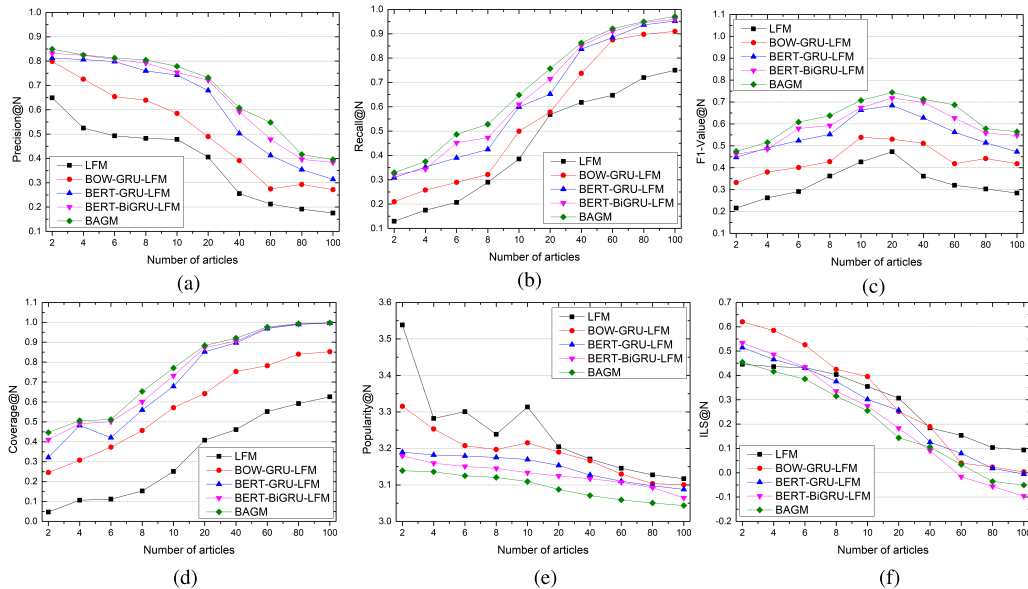
In this part, some variants of BAGM are presented as follows.

**LFM.** This is a matrix decomposition model of collaborative filtering. It does not consider the content information of the literature. First, a random factor vector is randomly initialized for the user and the literature in each matrix, and the internal product represents the predicted preference value. Then, by minimizing the mean square error of the predicted value and the actual value in the training set, the latent factor vectors of users and literature are continuously adjusted until the algorithm converges, and the extracted user latent factor matrix and literature latent factor matrix are saved. Finally, the preference prediction value is obtained through the inner product of trained latent factor vectors.  $N$  studies with the

<sup>2</sup><http://www.citeulike.org/faq/data.adp>

<sup>3</sup><https://www.librarything.com>





**FIGURE 3.** The performance of BAGM and 4 variants on CiteULike-a dataset in terms of Precision and so on by varying the number of recommended articles. (All reported improvements over baseline methods are statistically significant with  $p$ -value  $< 0.05$  based on the paired sample t-test.)

highest preference prediction value are included in the user recommendation list.

**BOW-GRU-LFM.** This model utilizes the bag-of-words model in the word embedding layer. First, each word in the title and abstract of the literature is expressed in the form of a single hot code. Each literature can be regarded as a high-dimensional sequence of word vectors. Then, the GRU network is stacked on the embedding layer. On the one hand, the dimension of the text feature vector is reduced by adjusting the number of network parameters. On the other hand, the memory capacity of the GRU network is applied to save more literature context information. Finally, the latent state sequence of the GRU network is regarded as the latent factor vector of the literature, and the inner product between the latent factor vector of the literature and the latent factor vector of the user is calculated in the same way as the LFM to generate recommendations for the user.

**BERT-GRU-LFM.** Unlike BOW-GRU-LFM, BERT-GRU-LFM first utilizes the case-insensitive BERT model to embed words of the literature content and map the words in the content to a 768-dimensional word vector. These word vectors contain the lexical, syntactic, and common semantic information of the words, and their performance is theoretically far superior to the bag-of-words model that cannot embody location information. Then, we utilized the BERT model to represent each article as a sequence of word vectors, which is passed as an input to the GRU layer, and the GRU network structure is used to semantically encode the text sequence. The output literature feature vector can not only reduce the dimension of the word vector but also capture the professional word semantics of the literature and optimize the common semantic encoding of the BERT model. Finally,

the literature feature vectors outputted by the GRU model are applied as the latent factor vectors of the literature, and a shallow LFM is used to generate a recommendation list for the user.

**BERT-BiGRU-LFM.** On the word embedding layer based on the BERT model, the BiGRU network structure is used to separately encode the words of the word vector sequence outputted by the BERT model from the forward direction and the reverse direction and obtain the two directions through matrix operations. The latent state of the combination is applied to obtain the literature latent feature vector for the calculation of the user preference prediction of the shallow LFM and generate recommendations for the user.

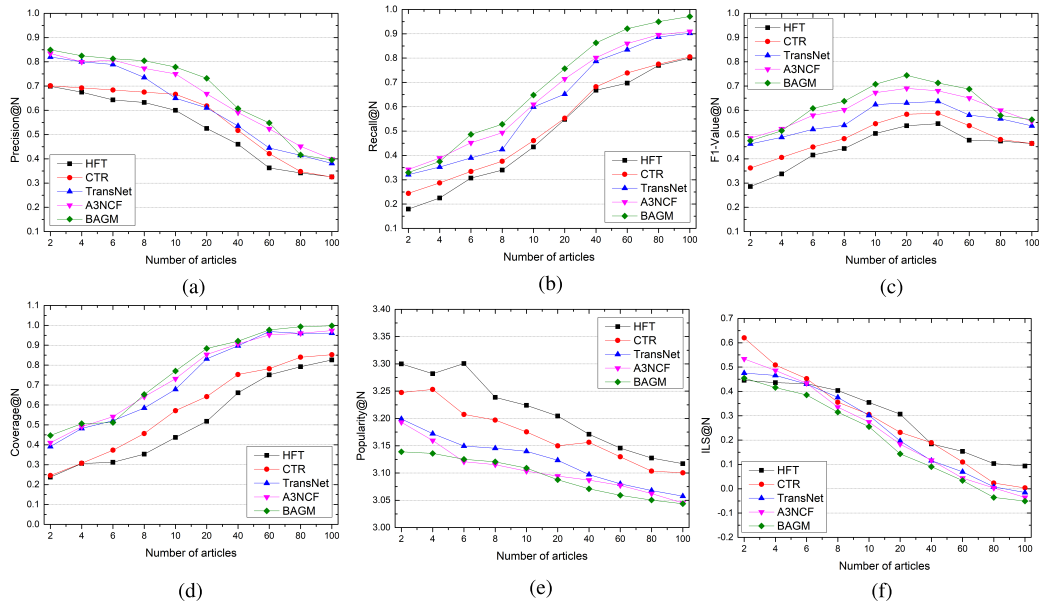
## 2) COMPETITORS

In this part, some methods we select as competitors are presented as follows.

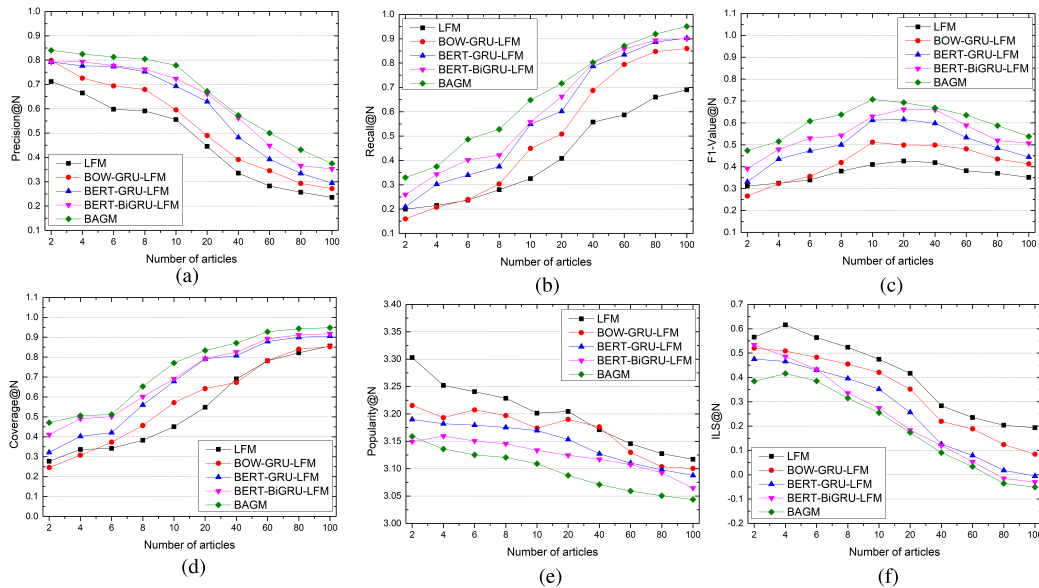
**HFT** [22]. This method models ratings and review texts with latent topic model. We apply it as a representative of the methods which apply an exponential transformation function to link the latent topics with latent factors. The topic distribution can be modeled on either users or items.

**CTR** [23]. This method also utilizes both review and rating information. It utilizes a topic model to learn the topic distribution of items, which is then applied as the latent factors of items in MF with an addition of a latent variable.

**TransNet** [46]. This method adopts a neural network framework for rating prediction. Reviews of users and items are passed into two CNNs respectively to learn the latent representations of users and items. The latent representations of a targeted user and a targeted item are concatenated and passed through a regression layer to estimate the rating.



**FIGURE 4.** The performance of BAGM and 4 competitors on CiteULike-a dataset in terms of Precision and so on by varying the number of recommended articles. (All reported improvements over baseline methods are statistically significant with p-value < 0.05 based on the paired sample t-test.)



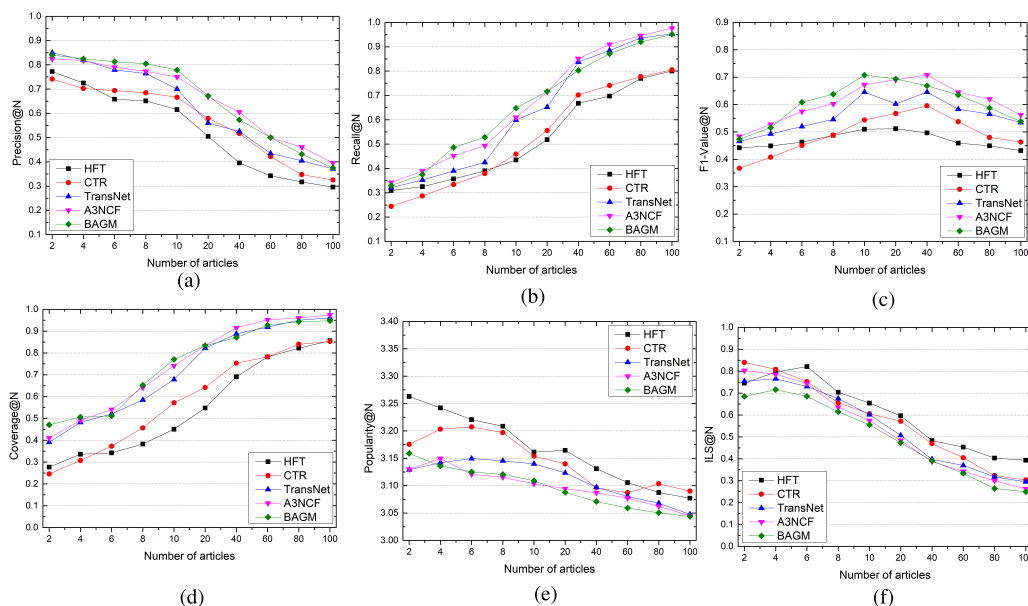
**FIGURE 5.** The performance of BAGM and 4 variants on LibraryThing dataset in terms of Precision and so on by varying the number of recommended articles. (All reported improvements over baseline methods are statistically significant with p-value < 0.05 based on the paired sample t-test.)

A3NCF [32]. This method uses review texts to guide the representation learning of users and items, and captures a user’s special attention on each aspect of the targeted item with an attention network.

**D. EXPERIMENTAL RESULTS AND ANALYSIS**

From the Fig. 3 and Fig. 5, we can observe: BAGM and its three variants outperform the LFM model on various evaluation metrics, which shows the hybrid recommendation

method using both text content information and user behavior data is better than a single collaborative filtering recommendation method. Deep mining of text content has certain practical significance for improving the performance of recommendation system for literature. As shown in the results on the ILS metric, BAGM performs well when the number of articles is small which shows that Algorithm 2 can effectively improve the diversity of recommendation results. The overall similarity between the literature will decrease



**FIGURE 6.** The performance of BAGM and 4 competitors on LibraryThing dataset in terms of Precision and so on by varying the number of recommended articles. (All reported improvements over baseline methods are statistically significant with  $p$ -value  $< 0.05$  based on the paired sample  $t$ -test.)

as the number of articles increases. In order to balance the diversity and precision of recommendation results, for low-activity users, priority is given to accuracy and ignoring its recommendation diversity. For high-activity users, attention is paid to the diversity of the recommendation results, which allows the precision of recommendation results to be certain loss.

As shown in Fig. 4 and Fig. 6, BAGM outperforms other competitors overall. The substantial improvement of our model over the baselines could be credited to four reasons: (1) The BERT model can employ a smaller literature content dataset to obtain more accurate and professional semantic representations of words to improve the accuracy of the recommendation results. (2) BiGRU network enhances the feature extraction capabilities of recurrent neural networks. (3) The user attention mechanism can make full use of the user latent factor vector to guide the weight distribution of the word vector sequence to improve the quality and interpretability of recommendation results. (4) Algorithm 2 assigns weights to literature according to user latent factor vector to increase the diversity of recommendation results.

## V. CONCLUSION

In this paper, we propose a hybrid neural network model using both researchers' behavior (favorites records) and literature content information to accurately represent the textual content information of the research papers and discover researchers' interest in achieving the recommended purpose. First, we utilize the BERT model to convert the one-hot encoding of words in the literature into word embedding vectors that contain certain semantic information. Then the

BiGRU is used to semantically encode words in the literature from forward and reverse, so that it can more reflect the context of the literature. The pooling technology based on user attention mechanism is applied to extract feature vectors which can accurately represent the content of the document text for subsequent recommendation tasks. Finally, a recommendation list is generated according to the user latent factor vector. We apply the real datasets to evaluate the proposed BAGM, the results show that our method has superiority and outperformance in comparison to the introduced baseline methods. In the future, we intend to use deep neural networks with better feature extraction capability to mine the full text of literature to extract more comprehensive literature feature vectors.

## REFERENCES

- [1] W. Zeng, A. Zeng, H. Liu, M.-S. Shang, and T. Zhou, "Uncovering the information core in recommender systems," *Sci. Rep.*, vol. 4, no. 1, p. 6140, May 2015.
- [2] M. Amami, G. Pasi, F. Stella, and R. Faiz, "An lda-based approach to scientific paper recommendation," in *Proc. Int. Conf. Appl. Natural Lang. Inf. Syst.*, Salford, U.K., Jun. 2016, pp. 200–210.
- [3] J. Beel, B. Gipp, S. Langer, and C. Breiter, "Research-paper recommender systems: A literature survey," *Int. J. Digit. Libraries*, vol. 17, no. 4, pp. 305–338, Nov. 2016.
- [4] S. Li, P. Brusilovsky, S. Su, and X. Cheng, "Conference paper recommendation for academic conferences," *IEEE Access*, vol. 6, pp. 17153–17164, 2018.
- [5] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.
- [6] C. Edwards, "Growing pains for deep learning," *Commun. ACM*, vol. 58, no. 7, pp. 14–16, Jun. 2015.

- [7] K. Song, M. Ji, S. Park, and I. Moon, "Hierarchical context enabled recurrent neural network for recommendation," in *Proc. 33rd AAAI Conf. Artif. Intell., AAAI, 31st Innov. Appl. Artif. Intell. Conf., IAAI, 9th AAAI Symp. Educ. Adv. Artif. Intell., EAAI*, Honolulu, HI, USA, Feb. 2019, pp. 4983–4991.
- [8] O. Foulds, L. Azzopardi, and M. Halvey, "Reflecting upon perceptual speed tests in information retrieval: Limitations, challenges, and recommendations," in *Proc. CHIIR Conf. Hum. Inf. Interact. Retr.*, Vancouver, BC, Canada, Mar. 2020, pp. 234–242.
- [9] X. Luo, M. Zhou, S. Li, Z. You, Y. Xia, and Q. Zhu, "A nonnegative latent factor model for large-scale sparse matrices in recommender systems via alternating direction method," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 3, pp. 579–592, Mar. 2016.
- [10] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for YouTube recommendations," in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 191–198.
- [11] S. Okura, Y. Tagami, S. Ono, and A. Tajima, "Embedding-based news recommendation for millions of users," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 1933–1942.
- [12] F. Cacheda, V. Carneiro, D. Fernández, and V. Formoso, "Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems," *ACM Trans. Web*, vol. 5, no. 1, pp. 1–33, Feb. 2011.
- [13] S. M. McNee, I. Albert, D. Cosley, P. Gopalkrishnan, S. K. Lam, A. M. Rashid, J. A. Konstan, and J. Riedl, "On the recommending of citations for research papers," in *Proc. ACM Conf. Comput. Supported Cooperat. Work CSCW*, 2002, pp. 116–125.
- [14] D. M. Pennock, E. Horvitz, S. Lawrence, and C. L. Giles, "Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach," 2013, *arXiv:1301.3885*. [Online]. Available: <https://arxiv.org/abs/1301.3885>
- [15] G. Adomavicius, B. Mobasher, F. Ricci, and A. Tuzhilin, "Context-aware recommender systems," *AI Mag.*, vol. 32, no. 3, pp. 67–80, 2011.
- [16] K. Sugiyama and M.-Y. Kan, "Scholarly paper recommendation via user's recent research interests," in *Proc. 10th Annu. Joint Conf. Digit. Libraries JCDL*, Jun. 2010, pp. 29–38.
- [17] B. Kazemi and A. Abhari, "A comparative study on content-based paper-to-paper recommendation approaches in scientific literature," in *Proc. 20th Commun. Netw. Symp.*, Baltimore MD, USA, 2017, pp. 1–10.
- [18] W. Waheed, M. Imran, B. Raza, A. K. Malik, and H. A. Khattak, "A hybrid approach toward research paper recommendation using centrality measures and author ranking," *IEEE Access*, vol. 7, pp. 33145–33158, 2019.
- [19] C. Basu, H. Hirsh, and W. W. Cohen, "Recommendation as classification: Using social and content-based information in recommendation," in *Proc. 15th Nat. Conf. Artif. Intell. 10th Innov. Appl. Artif. Intell. Conf., AAAI, IAAI*, Madison, WI, USA, Jul. 1998, pp. 714–720.
- [20] X. Dong, L. Yu, Z. Wu, Y. Sun, L. Yuan, and F. Zhang, "A hybrid collaborative filtering model with deep structure for recommender systems," in *Proc. 25th AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, Feb. 2017, pp. 1309–1315.
- [21] M. A. G. Pinto, R. Tanscheit, and M. Vellasco, "Hybrid recommendation system based on collaborative filtering and fuzzy numbers," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Jun. 2012, pp. 1–6.
- [22] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: Understanding rating dimensions with review text," in *Proc. 7th ACM Conf. Recommender Syst. RecSys*, 2013, pp. 165–172.
- [23] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining KDD*, 2011, pp. 448–456.
- [24] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 233–240.
- [25] B. Zhang, M. Zhu, M. Yu, D. Pu, and G. Feng, "Extreme residual connected convolution-based collaborative filtering for document context-aware rating prediction," *IEEE Access*, vol. 8, pp. 53604–53613, 2020.
- [26] Z. Cheng, Y. Ding, L. Zhu, and M. Kankanhalli, "Aspect-aware latent factor model: Rating prediction with ratings and reviews," in *Proc. World Wide Web Conf. World Wide Web WWW*, 2018, pp. 639–648.
- [27] Z. Cheng, X. Chang, L. Zhu, R. C. Kanjirathinkal, and M. S. Kankanhalli, "MMALFM: Explainable recommendation by leveraging reviews and images," *ACM Trans. Inf. Syst.*, vol. 37, no. 2, p. 16:1–16:28, 2019.
- [28] Y. Qu, H. Cai, K. Ren, W. Zhang, Y. Yu, Y. Wen, and J. Wang, "Product-based neural networks for user response prediction," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 1149–1154.
- [29] H. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah, "Wide & deep learning for recommender systems," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, Boston, MA, USA, Sep. 2016, pp. 7–10.
- [30] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "AutoRec: Autoencoders meet collaborative filtering," in *Proc. 24th Int. Conf. World Wide Web WWW Companion*, 2015, pp. 111–112.
- [31] H. Wang, N. Wang, and D. Yeung, "Collaborative deep learning for recommender systems," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Sydney, NSW, Australia, Aug. 2015, pp. 1235–1244.
- [32] Z. Cheng, Y. Ding, X. He, L. Zhu, X. Song, and M. Kankanhalli, "A<sup>3</sup>NCF: An adaptive aspect attention model for rating prediction," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3748–3754.
- [33] J. Y. Chin, K. Zhao, S. Joty, and G. Cong, "ANR: Aspect-based neural recommender," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 147–156.
- [34] T. Hofmann, "Latent semantic models for collaborative filtering," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 89–115, Jan. 2004.
- [35] T. Hofmann and J. Puzicha, "Latent class models for collaborative filtering," in *Proc. 16th Int. Joint Conf. Artif. Intell., IJCAI*, Stockholm, Sweden, vol. 1450, Aug. 1999, pp. 688–693.
- [36] T. Dai, T. Gao, L. Zhu, X. Cai, and S. Pan, "Low-rank and sparse matrix factorization for scientific paper recommendation in heterogeneous network," *IEEE Access*, vol. 6, pp. 59015–59030, 2018.
- [37] J. X. Chen, D. M. Jiang, and Y. N. Zhang, "A hierarchical bidirectional GRU model with attention for EEG-based emotion classification," *IEEE Access*, vol. 7, pp. 118530–118540, 2019.
- [38] W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang, "A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals," *Sensors*, vol. 17, no. 2, p. 425, Feb. 2017.
- [39] Y. Gong and Q. Zhang, "Hashtag recommendation using attention-based convolutional neural network," in *Proc. IJCAI*, Jul. 2016, pp. 2782–2788.
- [40] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 4945–4949.
- [41] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2015, pp. 577–585.
- [42] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, Jun. 2019, pp. 4171–4186.
- [43] Y. Zhu, R. Kiro, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 19–27.
- [44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent., ICLR*, San Diego, CA, USA, May 2015, pp. 1–15.
- [45] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [46] R. Catherine and W. Cohen, "TransNets: Learning to transform for recommendation," in *Proc. 11th ACM Conf. Recommender Syst.*, Aug. 2017, pp. 288–296.



**XU ZHAO** received the B.S. degree from the Prospecting and Engineering Technology Program (Applied Geophysics), Jilin University, Changchun, China, in 2017, where he is currently pursuing the master's degree in software engineering. His current research interests include data mining and natural language processing.



**HUI KANG** received the M.E. and Ph.D. degrees from Jilin University, in 1996 and 2007, respectively. She is currently an Associate Professor with the College of Computer Science and Technology, Jilin University. Her research interests include grid computing, information integration, and distributed computing.



**CHENKUN MENG** is currently pursuing the bachelor's degree with the College of Software, Jilin University, Changchun, China. His research interests include deep learning and natural language processing.



**TIE FENG** received the M.S. and Ph.D. degrees from Jilin University, in 1994 and 2007, respectively. He is currently an Associate Professor with the College of Computer Science and Technology, Jilin University. His research interests include software maintenance and evolution, reverse engineering, software refactoring, and automated software engineering.



**ZIQING NIE** is currently pursuing the bachelor's degree in computer science and technology with Jilin University, Changchun, China. Her research interests include deep learning and automated software engineering.

• • •