IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received September 3, 2018, accepted October 5, 2018, date of publication October 9, 2018, date of current version October 31, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2874959

Heterogeneous Knowledge-Based Attentive Neural Networks for Short-Term Music Recommendations

QIKA LIN[1], YAOQIANG NIU[2], YIFAN ZHU[1], HAO LU[1,3], KEITH ZVIKOMBORERO MUSHONGA[1], AND ZHENDONG NIU[1,4]

[1]School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China
[2]School of Computer Technology, Lanzhou Jiaotong University, Lanzhou 730000, China
[3]The State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[4]School of Computing and Information, University of Pittsburgh, Pittsburgh, PA 15260, USA

Corresponding author: Zhendong Niu (zniu@bit.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61370137 and in part by the Ministry of Education-China Mobile Research Foundation Project under Grant 2016/2-7.

ABSTRACT The current existing data in online music service platforms are heterogeneous, extensive, and disorganized. Finding an effective method to use these data in recommending appropriate music to users during a short-term session is a significant challenge. Another serious problem is that most of the data, in reality, obey the long-tailed distribution, which consequently leads to traditional music recommendation systems recommending a lot of popular music that users do not like on a specific occasion. To solve these problems, we propose a heterogeneous knowledge-based attentive neural network model for short-term music recommendations. First, we collect three types of data for modeling entities in user–music interaction network, i.e., graphic, textual, and visual data, and then embed them into high-dimensional spaces using the TransR, distributed memory version of paragraph vector, and variational autoencoder methods, respectively. The concatenation of these embedding results is an abstract representation of the entity. Based on this, a recurrent neural network with an attention mechanism is built, which is capable of obtaining users' preferences in the current session and consequently making recommendations. The experimental results show that our proposed approach outperforms the current state-of-the-art short-term music recommendation systems on one real-world dataset. In addition, it can also recommend more relatively unpopular songs compared with classic models.

INDEX TERMS Heterogeneous knowledge, data embedding, entity representation, attentive neural networks, short-term music recommendation.

I. INTRODUCTION

Currently, with the increasing popularity of online music streaming service platforms (MSSP) (e.g., Last.fm and Spotify), users can select from thousands of songs to listen to whenever they want and from wherever they are. On MSSP, there is massive data storing information such as songs, albums, singers, and their categories. At the same time, during the interaction between users and music platforms, a large amount of behavioral data is recorded, such as songs previously listened to, created or collected playlists, and marked favorites. As a result, there are many heterogeneous data on MSSP that are constantly increasing, which leads to huge challenges for music service in terms of describing user preferences and conducting song recommendations. Traditional music recommendation systems generally try to make long-term portraits and recommendations by using collaborative filtering (CF), content-based, or hybrid-based methods [1]–[10].

However, some psychology and sociology research on music shows that users' demands during a specific period are usually affected by their temporary context status [11]–[14]. For example, although people have their user preferences, the songs they listen to when they are happy and when they are down may be completely different. There is already some research focusing on the context-aware music recommender systems (CAMRSs), which are used for finding contextual patterns to better meet users' short-term needs. Existing CAMRSs have explored many types of context information, such as time [15], geographical location [16], weather [17], and mixed contexts [18]. The methods of these studies and many similar studies use probabilistic models, CF methods, or a combination of the two models [19]–[25].

In recent years, with the rapid development of deep learning, neural network models have been increasingly used in recommendation systems. The session-based recommendation is actually the same as the short-term music recommendation. Currently, the recurrent neural network (RNN) model with an attention mechanism provides the best solution [26]. The basic flow of the method is as follows: first, each item in the sequence is encoded by using a one-hot method and then input to an embedding layer for distributed representation of item information. Afterwards, an excellent model that uses a neural network with an attention mechanism can be obtained by learning and optimization. However, in real life, especially in industrialized datasets, the number of containing items is extremely large and usually exceeds 10^6. These data are subject to a long-tailed distribution. Coding with one-hot representation is obviously inefficient and time-consuming. Furthermore, such sparse data will lead to a lack of robustness and poor performance of the model. Nonetheless, ignoring most low-frequency items tends to result in a decrease in recommendation performance. Finding a way to balance these two issues is a very challenging but significant question. In this paper, we first use a large number of heterogeneous data to vectorize users and songs into dimensions of approximately 10^2. Then, embedding results are input into the RNN model to be optimized instead of one-hot representation, which greatly reduces the training complexity, and the results show that better performance has been achieved.

The main contributions of this article are as follows. 1) A music dataset is collected, which contains users' and songs' meta information (including textual descriptions and visual image data), graphic data (including associated artists, albums, playlists, tags, etc.), and interaction information between users and songs. Most of the data in our dataset is not available in previous datasets. 2) Multidimensional heterogeneous data are used to completely model users and songs, including user behavior data, song-related data, image data and text data. Thus, we can map entities to a high-dimensional space with dimensions of approximately 10^2. The experimental results show that this embedding method is very effective. 3) For the first time, the heterogeneous knowledge-based attentive neural network (HK-ANN) model is proposed and applied to music recommendation, which improves the input data and finally obtains state-of-the-art results. In addition, our model can also recommends unpopular music to a specific user.

II. RELATED WORK

A. TRADITIONAL MUSIC RECOMMENDATIONS

In recent years, with the gradual intensification of information and popularity of mobile devices, online music platforms have become indispensable in people's daily lives. It has become increasingly important to recommend the appropriate songs to users from a large number of songs. The traditional music recommendation model is usually applied to collaborative filtering (CF), content-based, or hybrid methods. Mcfee *et al.* [1] proposed a method for optimizing content-based similarity by learning from a sample of collaborative filter data, which used a variation on the typical bag-of-words approach for audio feature extraction. Dieleman and Schrauwen [2] processed sound information through the convolutional network to solve the cold start problem of the CF model. Wang and Wang [3] first used the deep belief network and probability graph model to learn audio information and make recommendations. Based on this, a hybrid model was used to automatically integrate the learned features and CF models. Chen *et al.* [4] performed sequence vectorization learning through a potential Markov embedding method. Aizenberg *et al.* [5] constructed a probabilistic CF recommendation model through music station playlists. Chang *et al.* [6] proposed a recommendation system based on personalized stress relief. In addition, some heterogeneous data was added to complete the information. Wang *et al.* [7] constructed heterogeneous networks between users, songs, and song metadata. Then they calculated similarity scores according to the weight of direct or indirect connections between nodes in the network to train the song's vector. Similarly, features were automatically extracted by building a network of relationships between entities, such as users and songs, and then recommendations were made in [8] and [9]. Chen *et al.* [10] added social factors to music recommendations. They linked the song-related ontology to form graphical data and calculated the similarity by using some transformations of graph theory. In general, the focus of these studies is shifting from single factor to making full use of large amounts of heterogeneous data and relational networks. However, these models are all designed to portray the user's overall preferences.

B. SHORT-TERM RECOMMENDATIONS AND THEIR APPLICATIONS

Verbert *et al.* [27] noted the importance of short-term recommendations. Similarity, Zhou *et al.* [28] reported two key observations, diversity and local activation, in the click-through rate (CTR) prediction scenario, which means that users are interested in different kinds of items, but only a part of users' historical behavior contributes to each click. Wang *et al.* [19], [20] first embedded songs using an ingenious method similar to word2vec [29], [30]. The user as a whole was represented as the mean of all song vectors of the user, and the context vector was expressed as the mean value of the song vector in the current context. The final recommendation result was determined by the total value of the sum of the dot product between the target song vector and these two vectors. Gupta *et al.* [21] decomposed a session into multiple subsessions and proposed a subsession based recommendation system by using songs' tags. Jin *et al.* [22] proposed a vectorization model to measure user portrait and context information. Users, songs, and context tags were vectorized using the Markov chain, and recommendations were made based on distance. Wang *et al.* [23] collected short-term information about users from mobile devices and constructed a probabilistic model to integrate short-term information and song content, which was used to make recommendations to meet the daily needs of users. Li and Liu [24] made recommendations through user behavior analysis and user sentiment extraction. Cheng and Shen [25] facilitated effective social music recommendations by considering users' location-related contexts as well as global music popularity trends.

Currently, the best performing methods for dealing with session-based recommendation problems are deep networks based on RNN [26], [31]. Hidasi *et al.* [26] applied RNN to session-based recommendations for the first time and designed special training and evaluation methods as well as ranking loss for this task. Tan *et al.* [32] used the gated recurrent unit (GRU) [33] and proposed two techniques, data enhancement and metering of input data transformations, to optimize the model. Adding item attribute information (such as text and image) into the RNN framework and associating several model frameworks with fused item attributes was explored by Hidasi *et al.* [34]. Bogina and Kuflik [35] considered the total time the user spends on the item in a temporary session. The longer the user interacts with items, the stronger the user's preference for it is in this session. They used two GRU encoders with an attention mechanism to model the overall preference as well as the main purpose from user behavior data. Then they integrated the two vector results together and calculated similarity with the candidate item vector through bilinear mode. The final score was calculated using the softmax layer. Quadrana *et al.* [36] proposed a hierarchical RNN model that describes the user's personal preference changes in the session and makes user personalized session recommendations. The user's historical information reflects the user's interests and is considered in the recommendation of the next session. In summary, the current use of RNN has made some progress in short-term recommendations, but its shortcomings that it only uses homogenous data, and one-hot representation is utilized as the input to the network, are still distinct. These problems make it unsuitable in the real world. As far as we know, our experiment is the first to apply an RNN model with an attention mechanism that uses many heterogeneous data for embedding to conduct short-term music recommendations.

III. METHODOLOGY

In this section, we introduce our proposed HK-ANN model to integrate a large number of heterogeneous data in online music service providers, including entity metadata, user behavior data and song related information. On this basis, we use the short-term feedback information of users and introduce recommendation methods to meet the short-term needs of users. In general, the entire process can be divided into entity embedding and short-term music recommendation. The overall architecture of our model is shown in Figure 1.

A. ENTITY EMBEDDING

There is a large amount of heterogeneous data in online music service platforms. Inspired by Zhang *et al.* [37], according to different categories, it can be roughly divided into graphic data, textual data, and visual data. The concatenation of these three data's embedding results represents the entity's embedding. Due to differences in data structures, we use different methods to process these data separately.

1) GRAPHIC EMBEDDING

Online music platforms have a large number of entities, which are interconnected to each other to form a very large graphic structure, such as the relationship between songs and albums, songs and singers, as well as playlists and songs. In addition, in the process of listening to songs, users will generate considerable interactive data by creating or collecting playlists and by marking songs they like. It is difficult to extract considerable useful information for graphic data using the traditional embedding method. Inspired by the idea of the knowledge graph, we use the TransR [38] method to vectorize the entities and relationships in the graph structure. TransR is the state-of-the-art approach for embedding heterogeneous networks. The general methods build entity and relation embeddings by regarding a relation as a translation from a head entity to a tail entity, which means these models simply put both entities and relations within the same semantic space. However, in reality, this is obviously unreasonable. TransR maps the entity and relationship of the model to a distinct space, i.e., entity space and multiple relation space and performs the translation in the corresponding relation space.

A simple illustration of TransR is shown in part (a) of Figure 1. For each relational triple $(\mathbf{h}, \mathbf{r}, \mathbf{t})$, entity embeddings are set as $h, t \in \mathbb{R}^k$ to represent the head entity and the tail entity, respectively, and relation embedding is set as $r \in \mathbb{R}^d$. Entities in the entity space are first projected into the relation space as h_r and t_r with operation $M_r \in \mathbb{R}^{k\times d}$, and then let $h_r + r \approx t_r$. The relation-specific projection can make head and tail entities that actually hold the relation close and distant from those that do not hold the relation [38].

The projected vectors of entities are defined as:

$$h_r = hM_r, t_r = tM_r \tag{1}$$

The corresponding score function is defined as follows:

$$f_r(h, t) = \|h_r + r - t_r\|_2^2 \tag{2}$$

By minimizing the score function above, we can obtain the vector representation of entities and relationships in the network at an abstract level. To increase generalization performance, some restrictions to embedding h, r, t and mapping matrices are needed, so we have $||h||_2 \leqslant 1$, $||r||_2 \leqslant 1$, $||t||_2 \leqslant 1$, $||hM_r||_2 \leqslant 1$, $||tM_r||_2 \leqslant 1$.

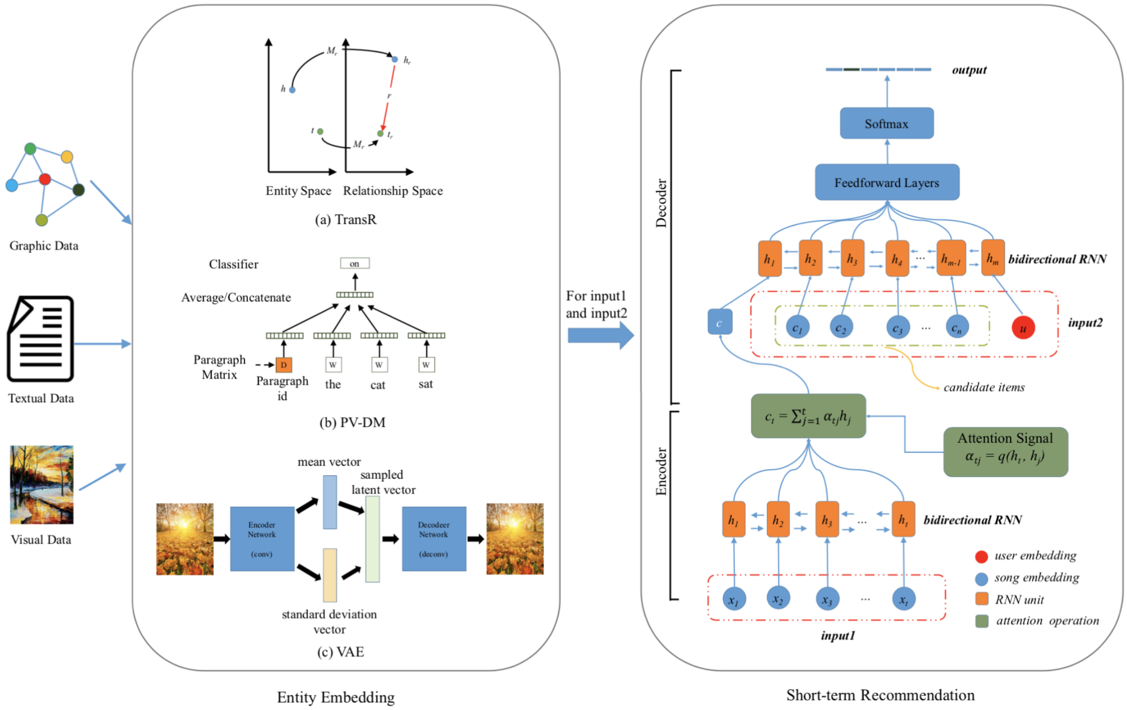


FIGURE 1. **The overall architecture of our HK-ANN model. It consists of entity embedding parts and short-term music recommendations.**

2) TEXTUAL EMBEDDING

In the music platform, there are many text description data, such as user self-description text and lyric content text, which have a significant impact on entity modeling. In this subsection, we investigate how to use a distributed memory model called the distributed memory version of paragraph vector (PV-DM) [39] to obtain the text information representation of an entity from text data. Similar to word2vec's idea [29], [30], PV-DM predicts the probability of the next word in a context through the average or concatenation of paragraph vector and word vectors. As shown in part (b) of Figure 1, every paragraph is mapped to a unique vector, represented by a column in matrix D and every word is also mapped to a unique vector, represented by a column in matrix W. Specifically, suppose that there are N paragraphs in the corpus, M words in the vocabulary, and we want to determine paragraph vectors such that each paragraph is mapped into p dimensions and each word is mapped to q dimensions, then the model has the total of $N \times p + M \times q$ parameters (excluding the softmax parameters). More formally, given a sequence of training words $w_1, w_2, w_3, \ldots, w_T$ and a paragraph d_i, the objective of the word vector model is to maximize the average log probability:

$$\frac{1}{T}\sum_{t=k}^{T-k} \log p(w_t|w_{t-k}, \ldots, w_{t+k}, d_i) \tag{3}$$

The prediction task is typically performed via a multiclass classifier, such as softmax. Here, we have:

$$p(w_t|w_{t-k}, \ldots, w_{t+k}, d_i) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}} \tag{4}$$

Each of y_i is an unnormalized log-probability for each output word i, computed as:

$$y = b + Uh(w_{t-k}, \ldots, w_{t+k}, d_i; W, D) \tag{5}$$

where U and b are the softmax parameters. h is constructed by concatenating or averaging the word vectors extracted from W and the paragraph vector extracted from D.

3) VISUAL EMBEDDING

In this subsection, we use an unsupervised deep learning approach, termed variational autoencoder (VAE) [40], [41], to extract the abstract representation of image information in multidimensional space. Unlike the usual stacked convolutional autoencoders (SCAE) [42], in addition to using the convolution in the process of encoding and deconvolution in decoding, VAE first generates a mean vector and a standard deviation vector when representing intermediate variables. Then, based on the trade-off of the Gaussian distribution between the two vectors, a new loss function is generated after adding the accuracy of generating the image. After a period of training, we can obtain a latent high-level feature representation with Gaussian constraint. The overall structure of the VAE is shown in part (c) in Figure 1.

We use cross-entropy (x_{ent}) to measure the difference between the original image (x) and the generated image ($\hat{x}$). The smaller x_{ent} is, the closer x is to $\hat{x}$.

$$x_{ent} = \sum_{i=1}^{n} -[x_i \cdot log\hat{x}_i + (1 - x_i) \cdot log(1 - \hat{x}_i)] \tag{6}$$

In addition, the encoder output mean (μ) and variance (σ^2) need to be constrained. Normally, the KL divergence is used to calculate the similarity between the two distributions.

$$KL = \frac{1}{2}(-\log\sigma^2 + \mu^2 + \sigma^2 - 1) \tag{7}$$

The total score function is:

$$Loss = x_{ent} + KL \tag{8}$$

B. SHORT-TERM MUSIC RECOMMENDATION

In the encoder and decoder stages, RNN is used to obtain the information in the sequence. In this paper, we use a bidirectional GRU as the basic unit. GRU gates essentially learn when and by how much to update the hidden state of the unit. The activation of the GRU is a linear interpolation between the previous activation h_{t-1} and the candidate activation $\hat{h}_t$:

$$h_t = (1 - z_t)h_{t-1} + z_t\hat{h}_t \tag{9}$$

where the update gate z_t is given by:

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \tag{10}$$

while the candidate activation function $\hat{h}_t$ is computed as:

$$\hat{h}_t = tanh(Wx_t + U(r_t \odot h_{t-1})) \tag{11}$$

and finally the reset gate r_t is given by:

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \tag{12}$$

The overall architecture of short-term music recommendations is shown on the right half of Figure 1. The bottom level's input (input1 in Figure 1) is the user's short-term sequence of listening songs which is represented by embedding results by using the method in the previous subsection. We utilize bidirectional GRU as the basic component of RNN in the music recommendation process. To capture the user's different preference degree in the short term, an item-level attention mechanism is adopted, which allows the decoder to dynamically select and linearly combine different parts of the input sequence like processing in machine translation [43]. This is shown as follows:

$$c_t = \sum_{j=1}^{t} \alpha_{tj} h_j \tag{13}$$

where α_{tj} models the matching degree between the j-th hidden state and the target at position t. All values of α show the relative importance of the item in the current input sequence and determine which one should be emphasized or ignored when making recommendations. It is computed as:

$$\alpha_{tj} = s(h_t, h_j) = v^T\sigma(Ah_t + Bh_j) \tag{14}$$

where the function s specifically computes the similarity between the final hidden state h_t and the representation of the previous item h_j. σ is an activate function such as sigmoid function, A is a transfer matrix for h_t, and B plays the same role for h_j.

TABLE 1. Entities and relations in the dataset.

No	Entity/Relation	Description	#count
1	U	User	8,832
2	S	Song	1,406,493
3	P	Playlist	99,960
4	Al	Album	420,789
5	Ar	Artist	151,013
6	T	Tag	74
7	$U \xrightarrow{like} S$	A user likes a song	1,445,737
8	$U \xrightarrow{create} P$	A user creates a playlist	57,045
9	$U \xrightarrow{collect} P$	A user collects a playlist	181,589
10	$Al \xrightarrow{belongs\ to} Ar$	An album belongs to an artist	425,857
11	$P \xrightarrow{include} S$	A playlist includes a song	9,751,358
12	$P \xrightarrow{is\ tag\ to} T$	A playlist is tagged to a tag	101,659
13	$Al \xrightarrow{include} S$	An album includes a song	1,406,493
14	$Ar \xrightarrow{perform} S$	An artist performs a song	1,586,789

Through the encoder with the attention mechanism, different items are assigned adaptive weights to represent the user's main purpose for the current session. In the decoding scheme, the representation of the current attention, the candidate items, and the representation in the current user are integrated as input to the network (input2 in Figure 1 are representations of candidate items and the current user). Through the bidirectional RNN and feedforward layers, the score of each candidate item is calculated. Finally, all scores are normalized through the softmax layer. Our model can be trained by using a standard mini-batch gradient descent on the cross-entropy loss:

$$Loss(p, q) = -\sum_{i=1}^{m} p_i \log(q_i) \tag{15}$$

where q is the prediction probability distribution and p is the actual distribution.

IV. EXPERIMENT AND RESULT

A. DATASET

The dataset of this paper is collected from NetEase Cloud Music,[1] which is one of the most popular music streaming platforms in China. As of Nov. 2017, NetEase Cloud Music has over 400 million users, 10 million songs, and the number of playlists that users created has increased to 400 million.[2] A large amount of complete heterogeneous data can be obtained on this platform, including entities, relationships and metadata.

As shown in Table 1, there are 6 types of entities and 8 types of relationships in the structured graph dataset. In the class of U-like S, considering the difference in the degree of liking a song, we divide this relationship into four subclasses according to the frequency of listening to songs. The dataset contains anomalous entities and relationships. The relationship between the entities can be seen more vividly in Figure 2,

[1] https://music.163.com/

[2] https://zh.wikipedia.org/wiki/网易云音乐

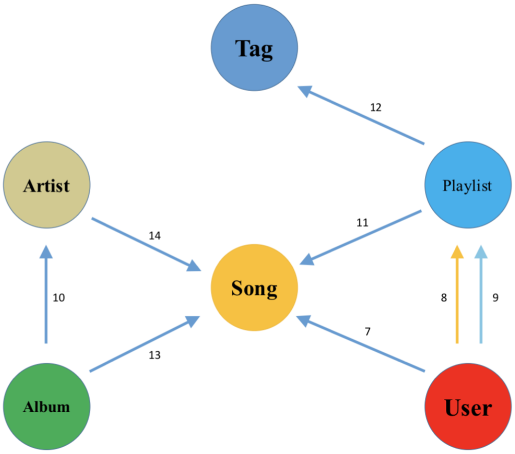


FIGURE 2. Graphic structure of the entities.

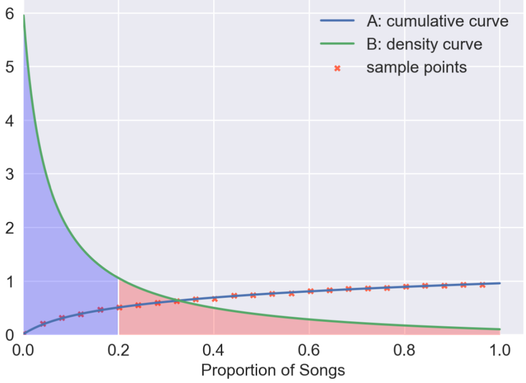


FIGURE 3. Cumulative curve and probability density curve for song proportion.

in which the number on the arrow corresponds to the number of the relation in Table 1. The number of songs was 10^7 and we found that they also obey the long-tailed distribution as shown in Figure 3. We first sort the songs from large to small according to the number of users who tag them as favorites. Curve A fits the cumulative proportion of users as the number of songs increases. Curve B is the probability density curve of curve A. From these two curves we can see that a small part of the popular music can meet the needs of most people, which is completely consistent with the long-tailed distribution.

User entities have their own set of avatars and self-description text. Each song also has a cover image for its corresponding album and lyrics text. These images and text data are stored and processed using the methods used in the previous section. The dataset is available online,[3] and it can be downloaded for free.

B. BASELINE METHODS

We compare the proposed HK-ANN model with two traditional methods and two of the latest methods for short-term music recommendations.

- *ItemKNN:* ItemKNN [44] is a traditional and popular collaborative filtering algorithm. The similarity between songs is measured by their cooccurrence frequency in the user's listening list.
- *FMPC:* Factorized personalized Markov chains (FPMC) [45] is a state-of-the-art hybrid model on the next-basket recommendation, which subsumes both a common Markov chain and the normal matrix factorization model. Instead of using the same transition matrix for all users, this method uses an individual transition matrix for each user, which results in a transition cube.
- *MEM:* The music embedding model (MEM) [19] first learns music pieces' embeddings by using music listening records and corresponding metadata. Then it infers and models users' global and contextual preferences for music from their listening records with the learned embeddings. Finally, it recommends appropriate music pieces according to the target user's preferences in order to satisfy her/his short-term requirements.
- *FGMSI:* A factor graphic model based on social influence (FGMSI) [10] exploits social influence in making music recommendations. It first constructs a heterogeneous social network and then calculates the meta path-based similarity. Finally, some specific features are extracted from the network and used for recommendations.

C. EVALUATION METRICS AND EXPERIMENTAL SETUP

The recommendation system is designed to recommend a list of potentially partial items to users. Its performance metrics measure the similarity between the recommendation results and the actual list. In this paper, four metrics are used to measure the recommendation performance of the model for the top-N recommendation problem.

- *Recall@N:* We set Recall@N as the primary metric, which represents the probability that an item a user likes is recommended. It is defined as the ratio between songs in the recommendation list and all songs that the user truly likes.
- *MRR@N:* Recall only measures whether the real item appears in the recommended list, but there is no factor to consider the rank, therefore we need to use other metrics. Mean reciprocal rank (MRR) is the average of reciprocal ranks of the desired items. The reciprocal rank is set to zero if the rank is larger than N or if the item is not in the recommended list.
- *NDCG@N:* Normalized discounted cumulative gain (NDCG) also measures the quality of the ranking list, which adds a position decay factor when calculating the cumulative gain. The calculation method is as follows:

$$DCG = \sum_{i=1}^{n} \frac{2^{r(i)-1}}{\log_2(1+i)} \tag{16}$$

$$NDCG = \frac{DCG}{IDCG} \tag{17}$$

[3] https://pan.baidu.com/s/1FwI6J0M8wA2DK-zMsYFqQw

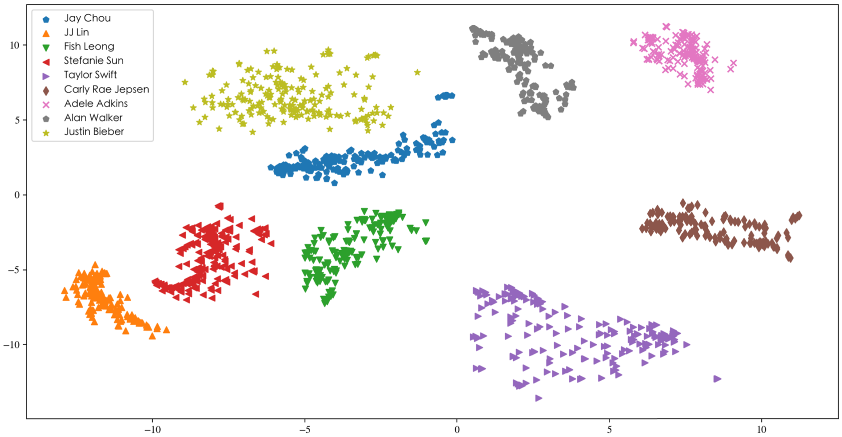


FIGURE 4. The embedding results of songs, which are categorized by nine popular singers.

where $r(i)$ represents the rank for item i in the true list and IDCG represents the DCG value in an ideal sorted state.

- *Novelty@N:* All of the above indicators only measure the similarity between the recommendation result and the true state, but it is worth noting that the popular items are usually given the higher priority in recommendation systems. For this reason, we used an unpopular metric named Novelty, which demonstrates the degree of an item being unknown to the user and an item being different from what the user has seen before [46]. Its' calculation approach is as follows:

$$p(i) = \frac{|\{u \in U, r_{ui} \neq \emptyset\}|}{|U|} \tag{18}$$

$$Novelty = \frac{\sum_i^n -\log_2 p(i)}{N} \tag{19}$$

where $p(i)$ is the fraction of users who are related to item i.

The proposed HK-ANN model uses 300-dimensional embeddings for songs and users by default. These embeddings are concatenated by 100-dimensional graphic embedding, textual embedding, and visual embedding. For the recurrent layers, we use GRU as it was found in [26] that they outperformed the long short-term memory (LSTM) [47] units. There are two GRU layers with 128 hidden units in encoder and decoder procedures. A dropout layer with 50% is added behind each GRU layer to increase the generalization and robustness of the model. When making a decoder, 100 candidate songs are input into the network and pass through bidirectional RNN layers, and there are two feedforward layers containing 512 and 256 hidden units. Optimization is performed using Adam [48] with the initial learning rate set to 0.001 and the mini-batch size fixed at 256. The model is defined and trained in Keras on a GeForce GTX 1080Ti GPU. The source code of our experiment is available online.[4]

D. EMBEDDING RESULT

To verify the effect of the embedding method, we chose all songs of some famous singers to visualize the vectorized representation of the song using the t-SNE method [49], which can map high-dimensional data to a low-dimensional space and then visualize it. We chose the songs of nine singers who are currently very popular around the world, including four Chinese singers (Jay Chou, JJ Lin, Fish Leong, and Stefanie Sun) and five foreign singers (Taylor Swift, Carly Rae Jepsen, Adele Adkins, Alan Walker, and Justin Bieber). The result is shown in Figure 4.

We can see that the songs of different singers are generally mapped to different positions in two-dimensional space. When comparing the overall distance between the songs of each singer, it can be seen that the Chinese singers are closer to each other than to the other singers. As for the Chinese female singers, the distance between Fish Leong and Stefanie Sun's songs is especially close as shown by the red and green dots in the figure. In reality, their songs and lyrics styles are similar to each other and their songs all tend to be youthful, positive and pure. As opposed to Chinese singers, the points of other singers' songs are scattered around the graph and the distance between them is generally large. This is due to the relatively large differences in their songs. All in all, songs can be mapped to high-dimensional vector spaces by using our embedding method. In this space, songs of the same artist are coherent, and different artist's songs are separated, which is exactly the embedding effect we want.

E. RESULT ANALYSIS

In this section, we will show the performance values of our model and compare them with the baseline models. In addition, we will explore the impact of different features on model performance.

1) COMPARISON WITH BASELINES

We performed experiments on the dataset using the four baselines mentioned above as well as the model we proposed.

[4] https://github.com/mistaken2/musicRec

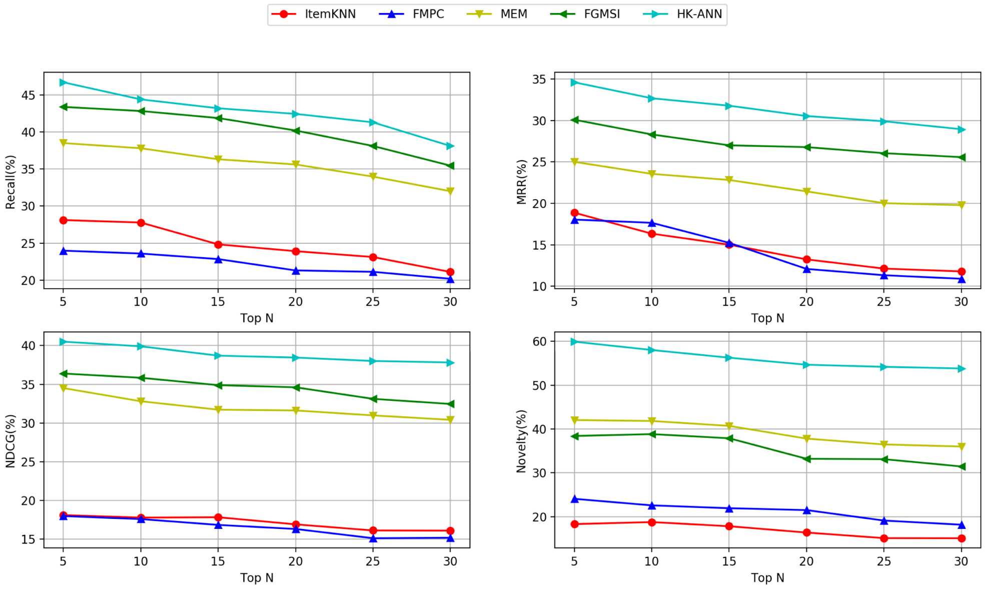


FIGURE 5. Our HK-ANN model's performance metrics compared with four baseline models. The horizontal axis indicates the number of recommendations per time.

TABLE 2. Different performance of five models for top-20 recommendaions.

Model	Recall@20	MRR@20	NDCG@20	Novelty@20
ItemKNN	0.2391	0.1323	0.1691	0.1639
FMPC	0.2131	0.1208	0.1631	0.2151
MEM	0.3562	0.2144	0.3162	0.3781
FGMSI	0.4021	0.2679	0.3461	0.3321
HK-ANN	**0.4245***	**0.3054***	**0.3845***	**0.5466***

The comparison of the results is shown in Figure 5 and Table 2.

As seen from the figure, the model we proposed achieved the best results in all four indicators. In the recall metric, the traditional ItemKNN and FMPC have the worst performance. They only achieved approximately 22% on the top-20 recommendations. This is because these two classic models only construct the user's overall portrait and do not consider their short-term factors. MEM and FGMSI achieved some improvement on their basis, especially the latter, which reached 35.62% and 40.21% on the top-20, respectively. The results of the MRR and NDCG indicators are similar to those of Recall, which indicates that the comparison results of each model do not change much, even if the ranking factor is added to the recommendation metric. However, in the Novelty indicator, it can be clearly seen that the four baselines are very poor. This is due to the long-tailed distribution of the songs. These models generally recommend popular songs to users to maintain accuracy. The HK-ANN model achieved a very large improvement compared to other models. This shows that our model recommends more unpopular songs to users with high accuracy. Another phenomenon is that as the number of recommendations increases, all indicators decline, which can be normally observed in most recommendation systems.

2) EFFECTS OF EMBEDDING DIMENSION

In our embedding process, there are three features in total, including graphic features, textual features and visual features. In order to explore the effect of the embedding dimension of different features on the recommended performance, we conducted this experiment. When the embedding dimension of a feature to be explored changes from 0 to 250, the dimensions of the other two features are set to a constant of 100. The experimental results are shown in Figure 6.

As the embedding dimension of a feature increases, the performance of the model increases and gradually stabilizes. When the dimension is less than 100, the change in the dimension corresponds to a greater impact on performance. These two phenomena show that as the dimension increases, the effect of embedding becomes increasingly better, and this trend gradually decreases. If we set an embedding dimension to 0, which means removing this feature, we can see that the performance of the model is greatly reduced. This indicates that each feature has a significant impact on recommendation results and cannot be ignored. Among the three indicators of Recall, MRR and NDCG, we can see that the change of graphic embedding has the greatest impact, and that the effect of visual embedding is the smallest. This means that the information in graphic embedding is the most significant for the recommendation, and the information contained in visual

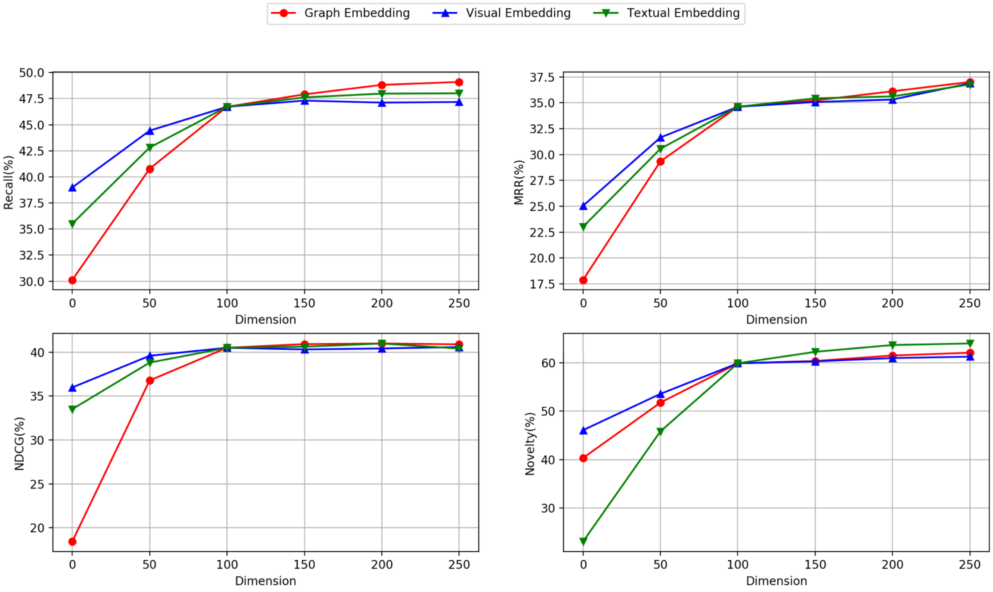


FIGURE 6. Effects of embedding dimension.

TABLE 3. Embedding impact in a recurrent neural network with an attention mechanism. Here, time consumed is expressed in minutes.

Model	Recall@20	MRR@20	NDCG@20	Novelty@20	#Time
TopSong-10000	0.1341	0.0803	0.0861	0.1694	25.44
TopSong-20000	0.1869	0.1318	0.1265	0.2189	34.82
TopSong-50000	0.2258	0.1622	0.1509	0.2337	45.26
TopSong-100000	0.2330	0.1863	0.1631	0.2882	72.90
HK-ANN	**0.4245***	**0.3054***	**0.3845***	**0.5466***	**15.8***

embedding is the least important. However, for the Novelty indicator, textual embedding plays the most important role, which shows that using textual information can increase the Novelty of the model recommendation results more remarkably compared to the other two features.

3) EMBEDDING IMPACT

We conduct an experiment to explore the impact of embedding on recommendation performance. Since there are 1,406,493 songs in our dataset, we cannot treat it directly as a collection without any processing. We use the approach commonly used in natural language processing (NLP) to process the word set as well as in the recommendation system to handle the item set. That is, take a subset that contains songs that most frequently listened to. The size of the collection varies from 10,000 to 100,000. The comparison between these models and our proposed HK-ANN model is shown in Table 3. In addition to the four evaluation metrics mentioned above, we also measure the time of each epoch in the training process.

We can see that as the length of the subset increases, the four performance indicators begin to slowly increase, which is accompanied by a rapid increase in training time. In the process of varying from 50,000 to 100,000, the Recall@20, MRR@20 and NDCG@20 of the model do not rise significantly, but the Novelty@20 and #Time are greatly improved. This indicates that as the size of the subset increases, the model not only recommends popular songs to users, but its time consumption becomes unacceptable. In contrast, our proposed method has a comprehensive advantage over the four performance metrics. The most important thing is that its time consumption is extraordinarily small, which is crucial for current online recommendation systems. For these reasons, it is expected to be widely used in large-scale industrial data.

V. CONCLUSION AND FUTURE WORK

A large amount of heterogeneous data on the Internet currently poses a great challenge for music recommendation systems, and effectively using these data becomes extremely important. This paper proposes an HK-ANN model to use heterogeneous data for embedding to conduct short-term music recommendations. First, we use TransR, PV-DM and VAE to embed graphics data, textual data and visual data respectively. The concatenation of these embedding results is a high-dimensional representation of the entity, which contains most of the heterogeneous information on online music platforms. Then, we propose an RNN model with an attention mechanism to obtain short-term preferences for the user listening to songs. The results show that our model has a better recommendation effect than the current mainstream music recommendation model. More importantly, since the actual data mostly obey the long-tailed distribution, the general recommendation system will mostly recommend popular items to users. Our model overcomes this shortcoming and can recommend songs that are relatively unpopular while also satisfying users' preferences.

Based on our experiments, there may be three directions that can be further explored in the future. First, we can use more data to describe the overall and short-term portrait of users, such as their opinions on social media and their comments on songs or singers. This information can improve the integrity of heterogeneous data and may have some positive influence on the improvement of short-term recommendation systems. Second, based on some existing research on the fusion of heterogeneous data, we will explore a variety of fusion methods and validate them through different datasets from multiple application scenarios. Third, we can apply this model to other areas, such as online education resource recommendations [50], [51]. We can model portraits of students, teachers and educational resources in the same way. According to students' current learning status, they are recommended to the corresponding resources or teachers. Teachers are also recommended to some students for learning from them. It is very practical and can help to make full use of educational resources, which is of great significance to online education.

ACKNOWLEDGMENT

The authors finally would like to thank the editor and all reviewers for their suggestions that have helped to improve the work.

REFERENCES

[1] B. McFee, L. Barrington, and G. Lanckriet, "Learning content similarity for music recommendation," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 8, pp. 2207–2218, Oct. 2012.

[2] A. van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 2643–2651.

[3] X. Wang and Y. Wang, "Improving content-based and hybrid music recommendation using deep learning," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 627–636.

[4] S. Chen, J. L. Moore, D. Turnbull, and T. Joachims, "Playlist prediction via metric embedding," in *Proc. ACM Knowl. Discovery Data Mining*, 2012, pp. 714–722.

[5] N. Aizenberg, Y. Koren, and O. Somekh, "Build your own music recommender by modeling Internet radio streams," in *Proc. Int. Conf. World Wide Web*, 2012, pp. 1–10.

[6] H.-Y. Chang, S.-C. Huang, and J.-H. Wu, "A personalized music recommendation system based on electroencephalography feedback," *Multimedia Tools Appl.*, vol. 76, no. 19, pp. 19523–19542, 2016.

[7] D. Wang, G. Xu, and S. Deng, "Music recommendation via heterogeneous information graph embedding," in *Proc. Int. Joint Conf. Neural Netw.*, 2017, pp. 596–603.

[8] C. Guo and X. Liu, "Dynamic feature generation and selection on heterogeneous graph for music recommendation," in *Proc. IEEE Int. Conf. Big Data*, Dec. 2016, pp. 656–665.

[9] C. Guo and X. Liu, "Automatic feature generation on heterogeneous graph for music recommendation," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2015, pp. 807–810.

[10] J. Chen, P. Ying, and M. Zou, "Improving music recommendation by incorporating social influence," *Multimedia Tools Appl.*, vol. 4, no. 1, pp. 1–21, 2018.

[11] J. H. Lee and J. S. Downie, "Survey of music information needs, uses, and seeking behaviours: Preliminary findings," in *Proc. ISMIR*, 2004, pp. 441–446.

[12] M. Kaminskas and F. Ricci, "Contextual music information retrieval and recommendation: State of the art and challenges," *Comput. Sci. Rev.*, vol. 6, nos. 2–3, pp. 89–119, 2012.

[13] M. Braunhofer, M. Kaminskas, and F. Ricci, "Recommending music for places of interest in a mobile travel guide," in *Proc. ACM Conf. Recommender Syst.*, 2011, pp. 253–256.

[14] L. Baltrunas *et al.*, "InCarMusic: Context-aware music recommendations in a car," in *Proc. E-Commerce Web Technol.-Int. Conf. (EC-Web)*, Toulouse, France, Aug./Sep. 2011, pp. 89–100.

[15] R. Dias and M. J. Fonseca, "Improving music recommendation in session-based collaborative filtering by using temporal context," in *Proc. IEEE Int. Conf. TOOLS Artif. Intell.*, Nov. 2013, pp. 783–788.

[16] M. Kaminskas, F. Ricci, and M. Schedl, "Location-aware music recommendation using auto-tagging and hybrid matching," in *Proc. ACM Conf. Recommender Syst.*, 2013, pp. 17–24.

[17] H.-S. Park, J.-O. Yoo, and S.-B. Cho, "A context-aware music recommendation system using fuzzy Bayesian networks with utility theory," in *Proc. Int. Conf. Fuzzy Syst. Knowl. Discovery*, 2006, pp. 970–979.

[18] W. Yao, J. He, G. Huang, J. Cao, and Y. Zhang, "A graph-based model for context-aware recommendation using implicit feedback data," *World Wide Web-Internet Web Inf. Syst.*, vol. 18, no. 5, pp. 1351–1371, 2015.

[19] D. Wang, S. Deng, X. Zhang, and G. Xu, "Learning to embed music and metadata for context-aware music recommendation," *World Wide Web-Internet Web Inf. Syst.*, vol. 21, no. 5, pp. 1399–1423, 2018.

[20] D. Wang, S. Deng, and G. Xu, "Sequence-based context-aware music recommendation," *Inf. Retr. J.*, vol. 21, nos. 2–3, pp. 230–252, 2018.

[21] K. Gupta, N. Sachdeva, and V. Pudi, "Explicit modelling of the implicit short term user preferences for music recommendation," in *Proc. Eur. Conf. Inf. Retr.*, 2018, pp. 333–344.

[22] L. Jin, D. Yuan, and H. Zhang, "Music recommendation based on embedding model with user preference and context," in *Proc. IEEE Int. Conf. Big Data Anal.*, Mar. 2017, pp. 688–692.

[23] X. Wang, D. Rosenblum, and Y. Wang, "Context-aware mobile music recommendation for daily activities," in *Proc. ACM Multimedia*, 2012, pp. 99–108.

[24] Q. Li and D. Liu, "Research of music recommendation system based on user behavior analysis and word2vec user emotion extraction," in *Proc. Int. Conf. Intell. Interact. Syst. Appl.*, 2017, pp. 469–475.

[25] Z. Cheng and J. Shen, "Just-for-Me: An adaptive personalization system for location-aware social music recommendation," in *Proc. Int. Conf. Multimedia Retr.*, 2014, pp. 185–192.

[26] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk. (2015). "Session-based recommendations with recurrent neural networks." [Online]. Available: https://arxiv.org/abs/1511.06939

[27] K. Verbert *et al.*, "Context-aware recommender systems for learning: A survey and future challenges," *IEEE Trans. Learn. Technol.*, vol. 5, no. 4, pp. 318–335, Oct./Dec. 2012.

[28] G. Zhou *et al.* (2017). "Deep interest network for click-through rate prediction." [Online]. Available: https://arxiv.org/abs/1706.06978

[29] T. Mikolov, K. Chen, G. Corrado, and J. Dean. (2013). "Efficient estimation of word representations in vector space." [Online]. Available: https://arxiv.org/abs/1301.3781

[30] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[31] A. Singhal, P. Sinha, and R. Pant, "Use of deep learning in modern recommendation system: A summary of recent works," *Int. J. Comput. Appl.*, vol. 180, no. 7, pp. 17–22, 2017.

[32] Y. K. Tan, X. Xu, and Y. Liu, "Improved recurrent neural networks for session-based recommendations," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, 2016, pp. 17–22.

[33] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. (2014). "Empirical evaluation of gated recurrent neural networks on sequence modeling." [Online]. Available: https://arxiv.org/abs/1412.3555

[34] B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk, "Parallel recurrent neural network architectures for feature-rich session-based recommendations," in *Proc. ACM Conf. Recommender Syst.*, 2016, pp. 241–248.

[35] V. Bogina and T. Kuflik, "Incorporating dwell time in session-based recommendations with recurrent neural networks," in *Proc. ACM Conf. Recommender Syst.*, 2017, pp. 57–59.

[36] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, "Personalizing session-based recommendations with hierarchical recurrent neural networks," in *Proc. ACM Conf. Recommender Syst.*, 2017, pp. 130–137.

[37] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 353–362.

[38] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 2181–2187.

[39] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1188–1196.

[40] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–14.

[41] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1278–1286.

[42] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Proc. Int. Conf. Artif. Neural Netw.*, 2011, pp. 52–59.

[43] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

[44] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan./Feb. 2003.

[45] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized Markov chains for next-basket recommendation," in *Proc. Int. Conf. World Wide Web*, 2010, pp. 811–820.

[46] M. Kaminskas and D. Bridge, "Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems," *ACM Trans. Interact. Intell. Syst.*, vol. 7, no. 1, pp. 1–42, 2016.

[47] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

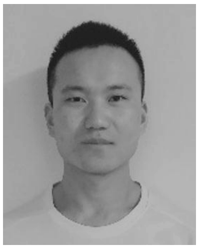
[48] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

[49] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[50] S. Wan and Z. Niu, "A learner oriented learning recommendation approach based on mixed concept mapping and immune algorithm," *Knowl.-Based Syst.*, vol. 103, pp. 28–40, Jul. 2016.

[51] J. K. Tarus, Z. Niu, and A. Yousif, "A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining," *Future Gener. Comput. Syst.*, vol. 72, pp. 37–48, Jul. 2017.

QIKA LIN received the B.E. degree in chemical engineering and technology from the Beijing Institute of Technology, Beijing, China, in 2016, where he is currently pursuing the M.S. degree with the School of Computer Science and Technology. His research interests include intelligent e-learning, data mining, and natural language processing.



YAOQIANG NIU received the B.E. degree in computer science and technology from the University of Shanghai for Science and Technology, Shanghai, China, in 2017. He is currently pursuing the M.S. degree with the School of Computer Technology, Lanzhou Jiaotong University, Lanzhou, China. His research interests include intelligent computing, machine learning, and data mining.

YIFAN ZHU received the B.E. degree in computer science from Beijing Information Science and Technology University, Beijing, China, in 2016. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing. His research interests include opinion mining, user profiling, and social computing.

HAO LU received the M.Sc. degree from the School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China, in 2009. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing.

He is currently an associate professor with The State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing. His current research interests include knowledge automation, social computing, and web intelligent.

KEITH ZVIKOMBORERO MUSHONGA received the B.A. degree in French, with minors in Spanish and English, from Winthrop University, USA, in 2016. He is currently pursuing the master's degree in computer science with the Beijing Institute of Technology. His research interests include intelligent language tutoring systems, Web media analysis, and computational linguistics.

ZHENDONG NIU received the Ph.D. degree in computer science from the Beijing Institute of Technology, Beijing, China, in 1995. From 1996 to 1998, he was a Post-Doctoral Researcher with the University of Pittsburgh, Pittsburgh, PA, USA, where he has been a Joint Professor with the School of Computing and Information since 2006. He was a Research/Adjunct Faculty Member with Carnegie Mellon University, Pittsburgh, from 1999 to 2004. He is currently a Professor and the Deputy Dean with the School of Computer Science and Technology, Beijing Institute of Technology. His current research interests include informational retrieval, software architecture, digital libraries, and Web-based learning techniques.

Dr. Niu was a recipient of the IBM Faculty Innovation Award in 2005 and the New Century Excellent Talents in University of Ministry of Education of China in 2006.