2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications

Hierarchical Attention Based Recurrent Neural Network Framework For Mobile MOBA Game Recommender Systems

Qiongjie Yao, Xiaofei Liao, and Hai Jin
Services Computing Technology and System Lab
Cluster and Grid Computing Lab
School of Computer Science and Technology
Huazhong University of Science and Technology, Wuhan, 430074, China
Email: xfliao, hjin@hust.edu.cn

***Abstract*—The mobile *multiplayer online battle arena* (MOBA) game is a genre of real-time strategy video games on mobile devices, such as King of Glory. The main business model is to drive players to purchase items like heroes or skins. Recommending items based on player interest is the core task of recommender systems. In the MOBA game, player interest changes over the game experience, which is implied in player behavior based on historical game matches. Match sequences, that consist of every match in the timeline, indicate how players interact with the game and the change process of player interest. *Recurrent neural networks* (RNNs) are employed by many recommendation scenes to model sequence data to profile user preference for better recommendation accuracy. However, their RNNs based frameworks ignore the interpretability of recommendation results, which is an important requirement for mobile MOBA games.**

To solve this challenge, we propose an interpretable RNN framework based on hierarchical attention in this work, which is inspired by the attention mechanism applied in machine translation. The main component *long short-term memory* (LSTM), that is the RNN variant, models player interest from historical match sequences, and the hierarchical attention is used to measure the effect factors of matches and behavior events happened in a match. To verify effectiveness, we train several models on real mobile MOBA game King of Glory datasets. Compared to non-sequence models, our model achieves 2% higher accuracy; with hierarchical attention, the proposed model can interpret the recommendation results effectively compared to naive RNN based models.

***Index Terms*—Mobile MOBA game, Player interest, LSTM, Hierarchical attention**

I. Introduction

Mobile *multiplayer online battle arena* (MOBA) games are a kind of mobile game, which is a fusion of action games, role playing games, and real-time strategy games. At the beginning of the game, players are divided into two teams, and every player chooses a single role, generally called a hero, to fight. When a team destroys the main building of the other team, the game is over; players can continue to the next game with the same mechanism. However, every match brings different game experience because of different players and heroes. The better game experience motivates the game interest and purchase preference of players, e.g., killing more enemies, winning the *most valuable player* (MVP) title, or winning the game. Players prefer to purchase items that hold their interest. In the MOBA game, the main items are heroes and skins that have great benefits to game companies. Using King of Glory as an example, approximately 50 million players are active and consume roughly RMB 100 million daily according to the first quarter Tencent earnings in 2017. Therefore, precise item recommendations are a challenging and valuable task for mobile MOBA games.

For precise recommendations, the critical problem is to model player interest. Some studies [1], [2] assume that the player interest is stable or changes slowly. The player profile information is used such as age, gender, or favorite cartoon style. This assumption seems to be unrealistic for mobile games because player interest changes with game experience. Many new players hope to show their excellent game talent, so they prefer to play powerful heroes that kill enemies easily. After winning several matches, they prefer to play heroes with complex operations because these heroes are more challenging. After losing matches, they prefer to play skilled heroes and cooperate well with other players to win. Thus, a model to profile the dynamic interest of players is needed in our work. *Collaborative filtering* (CF) [3] is based on the idea that similar users typical have similar interest. Player purchase records are collected to compute the similarity of players or items. However, purchase events in mobile MOBA games are sparse compared with play events. Before purchasing items, a player usually plays many matches, even hundreds or thousands of matches. The player interest is timely and shifts over these matches. When a player plays a hero and wins the game as the MVP, he or she is willing to purchase items related to that hero. With more matches, this purchase preference will be augmented or attenuated based on new game experience. Obviously, CF methods based on purchase events are stale to model dynamic player interest.

Our work overcomes this shortcoming and explores the recent player behaviors to model timely and dynamic interest of players. Recent player behaviors are represented as a match sequence, which consists of every match over time. Moreover,

978-1-7281-1141-4/18/$31.00 ©2018 IEEE
DOI 10.1109/BDCloud.2018.00060

many events of player behaviors are chosen to represent each match, not just a single event like with a hero *identification* (ID).

Modeling user interest based on historical behavior logs has been widely studied by many recommendation scenes; early work about personalized news recommendation [4] determined that the current user interest is critical for recommendation systems and developed a Bayesian framework to model user news interest based on historical click activities. YouTube recommendations [5] average the historical video watch sequences and search tokens sequences with a watch vector and a search vector. However, this averaging approach ignores the order of the watch and search tokens. Thus, a better model is needed to exploit the temporal dependency of historical behavior sequences. Following this idea, click prediction [6] employs an RNN model to extract the vector representation of user interest from historical click sequences, which is the first RNN-based recommendation framework. The experiments demonstrate that the RNN model can significantly improve the click prediction accuracy compared with the *logistic regression* (LR) and *deep neural network* (DNN) models. After that, the RNN models are used in more recommendation scenes, e.g., personal medical diagnosis [7] [8], E-commerce recommendations [9], [10], [11], and news recommendations [12].

To our best knowledge, this is the first study to extend RNNs to mobile games by profiling player interest based on recent match sequences. However, it is difficult with the single RNN model to interpret the recommendation results due to the deep neural network characteristics. For MOBA games, the interpretability of recommendation results is an actual requirement. The suitable recommendation reason not only is helpful to guide players to purchase items but also provides player feedback about the current game version, which is important for developers. Developers hope to deeply understand player interest and design new heroes and skins in future game updates. Thus, developers focus on the behavior events that interest players to make purchases. Inspired by machine translation [13], it introduces an attention mechanism for word alignment where words in source sentence are assigned different weights to make up an encoding vector, not just the last hidden state [14]. However, this simple attention-based RNN framework cannot be extended to recommendation of MOBA games because every state of the match sequence includes rich player behavior events.

In this paper, an interpretable *long short-term memory* (LSTM) based recommendation framework is built by exploiting recent player behavior. The LSTM based component is used to model player interest from long match sequence. Each match is represented by the weighted sum of all event vectors where an event is encoded separately by an embedding network. To achieve interpretability, a hierarchical attention is designed to include the match attention module and the event attention module. The match attention measures the importance of each match in a match sequence, and the event attention measures the importance of each event in a match. The importance indicates the recommendation reason. Finally, the proposed framework is evaluated on real game data of King of Glory, which is currently the top grossing mobile MOBA game. Compared to DNN which only uses the last match as input, the RNN models achieve 58% top-5 accuracy, which is 2% higher accuracy than DNN models.

By analyzing the statistics of the attention values, the impact of player historical behavior on future purchase follows the principle of the recency effect and the impact of the primacy effect is weak. The more recent matches provide more information for purchase prediction than older historical matches. The proposed framework provides a quantifiable approach for revealing the relation between historical behavior and future behavior. It is flexible to be extended to other scenarios without considering many prior assumptions of recommendation problems. Furthermore, this framework can be used to identify important behavior features. For the set of behavior features in this work, the hero ID and hero types dominate purchase decisions because these strong events indicate subjective wishes of players. This observation is consistent with the fact that many studies employ items ID as their important feature [9], [11]. It proves indirectly that this quantitative approach is effective, and then we can mine useful information from other behavior events based on the importance. In a word, this hierarchical attention based framework provides a flexible and promising approach to interpret the recommendation results.

The remainder of this work is organized as follows. Section II introduces the LSTM models and the attention mechanism. Section III summarizes the related work. Section IV mainly details the basic recommendation model and the proposed framework. Section V evaluates the proposed framework. Finally, the conclusion is presented in Section VI

II. Background

A. Basic LSTM architecture

RNNs [16], [17] are a type of deep neural networks that fold in the time axis. They are employed to abstract feature vectors of sequential data where the hidden activations at one time are computed by combining the hidden activations at a previous time and the current input. However, vanilla RNNs cause the gradient vanishing problem when training long sequences. In order to solve this problem, some works design many variants of RNNs. LSTM architectures are a typical variant, which memorizes long sequence information by the input gate, the output gate, and the forget gate. Here, we adopt the LSTM architecture defined by previous work [15].

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{1}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{2}$$

$$c_t = f_t c_{t-1} + i_t tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{3}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \tag{4}$$

$$h_t = o_t tanh(c_t) \tag{5}$$

where i_t, f_t, and o_t are the input gate, forget gate, and output gate at time t. x_t, c_t, and h_t represent the input, the cell activation vectors, and the hidden activations at time t respectively. σ is the logistic sigmoid function used for nonlinear transformation. W and b are corresponding weights and biases. Compared to the standard RNN model, LSTMs have more trained parameters; however, the design of the three gates makes LSTMs achieve better prediction performance [8].

B. Attention mechanism

The attention mechanism is successfully used in neural machine translation [13], [18]. For a sentence $x = (x_1, ..., x_T)$ where x_t is the t-th word, it is encoded by the RNN model to a context vector c by following formulas in encoder-decoder framework.

$$h_t = \sigma(x_{t,h_{t-1}}) \tag{6}$$

$$c = \varphi(h_1, ..., h_t) \tag{7}$$

h_t is a hidden state at time t, and c is the encoder vector which is the vector representation of the sentence. σ is a nonlinear function and φ is the compounded function. The simplest function for all hidden states h is $c = h_t$. Then, the prediction of target sentence is computed by the below formula.

$$p(y_t|y_1, ..., y_{t-1}, c) = g(y_{t-1}, s_t, c) \tag{8}$$

However, this c does not consider the word alignment between the source sentence and the target sentence. By introducing an attention mechanism the words in the source sentence are assigned a weight a_{ij}. Then the context c_i is computed by (9)

$$c_i = \Sigma_{j=1}^{S} a_{ij} h_j \tag{9}$$

$$p(y_i|y_1, ..., y_{i-1}, x) = g(y_{i-1}, s_i, c_i) \tag{10}$$

and the probability of the ith word is predicted by (10). The attention is used for word alignment between the source sentence and the target sentence.

III. Related Work

RNN-based models have been extended to many scenes. Previous work [6] first introduces an RNN-based framework because user behavior on advertisement can be predicted by exploiting the temporal relevant events in historical behavior. Personalized recommendations [11] use a deep RNN framework to mine browsing patterns of users and provide a real-time recommendation in e-commerce systems. Their experiments on the Kaola testing system show that RNN models achieve an order of magnitude better result than previous recommendation technique such as CF. Session-based recommendations [9] train an RNN model with a ranking loss only used item-IDs. To use more features, other studies [10] introduce a number of parallel RNN architectures to model sessions based on clicks and features of clicked items.

The attention mechanism was proposed for machine translation [13]. In the encoder-decoder framework the attention mechanism is applied to select relevant words in source sentences for words in target sentences before translation. Some papers have tried to use attention to interpret the prediction result. Retain [20] is an interpretive predictive model that mimics real medical processes, so a reverse time attention mechanism was designed in an RNN for the prediction task. However, this paper does not consider long medical information. Dipole [21] models the information of both past visits and future visits by employing *bidirectional recurrent neural networks* (Bi-RNN). They introduce three attention mechanisms to measure the relationships of different visits for making predictions. However, it is a single layer attention network, and must employ ReLU as the activation function to obtain an interpretive result. Our work overcomes these limitation by designing hierarchical attention, which is flexible to recommendation reason and scalable to new player behavior.

IV. Hierarchical Attention-Based Recurrent Neural Network Framework

Due to the recurrent network structure, which is different from feedforward neural network structure, RNNs have been proven to model temporal dependencies in sequences effectively. For a new scene, one needs to design a specific RNNs-based framework to exploit the characteristic of the scene. In this section the recommendation problem is defined firstly, and samples are constructed; then, a typical RNNs based framework is introduced for a recommender system. Finally, the proposed hierarchical attention based recurrent neural network framework is detailed.

A. Formulation of the recommendation problem

The first step of building a recommender system is to understand the recommendation scene and abstract the basic mathematical model. The model determines the learning method and sample structure. Search-based online advertising [19] attempts to maximize the similarity of the query and advertisements for recommendation and constructs a pair of queries and advertisements as a sample. Unlike these, mobile game recommendation scene is that the game shop recommends top-k items for players based on current player interest after a game is over. The recommendation task becomes the prediction of the probability of a player purchasing some items in given game context, and the formula for the recommendation task is as follows.

$$p(y = j|x) = \frac{e^{w_j^T x}}{\sum_{n=1}^{N} e^{w_n^T x}} \tag{11}$$

where x is the vector representation of player interest, and y is the items that players purchase. According to this definition, the precise vector representation of player interest is critical for recommendation performance. This work models player interest from the temporal player behavior among matches. Following this idea, training samples are constructed for the proposed model.

B. Sample structure

We construct training and test samples from logs databases of King of Glory, a popular mobile MOBA game. A sample is represented as a (context, label) tuple. The context is a match sequence ordered by timestamps in the form of $\{m_0, m_1, ..., m_{n-1}\}$, where m_t is the vector representation of the t-th match in sequence and $t \in [0, n-1]$. The label is items such as heroes or skins purchased by players after the match m_{n-1}. A simple approach to represent each match uses the hero ID. In this training data, there are sixty-eight heroes; thus m_t is a sixty-eight dimension vector with one-hot encoding. This representation method is similar to the studies [9], [11] where they only use web pages ID or items ID to represent each element in sequence.

Actually, mobile MOBA games are a complex and deep interactive system. The hero ID sequence is not enough to extract player interest. Every match contains some player behavior, which is called events such as hero ID, hero types, MVP state, winning state. Some of these behavior events bring players good game experience and then contribute the player interest. Therefore, we represent m_t as a combination $(e_{t,0}, e_{t,1}, ..., e_{t,k})$, where $e_{t,j}$ is the j-th event in the t-th match and $j \in [0, k]$.

However, there is a challenging problem about setting the proper n for sequences. A fixed n only considers the logistical order of matches in sequence and ignores the real time interval between two matches. In fact, the frequency of game play for everyone is different, and the time interval can cross days, weeks or even months, although intervals are adjacent in logic. It is difficult for a player to remember the match detail clearly when played at long ago or even yesterday. Recent matches happened in a short real time are useful information to model player interest. In an e-commerce system [11], they group web pages accessed by users in time interval T into a user session. T is a predefined threshold, e.g., 30 minutes. Their n is four where three states represent web pages accessed recently, and a history state represents compressed historical information. However, this short sequence loses some useful player interest information.

In our work, more historical information is exploited in each session. A match in a mobile MOBA game lasts an average of 15 to 20 minutes. Small thresholds make the match sequence short and provide less information. Considering the characteristics of MOBA games, we organize the matches in a day into a player session and feed it into the RNN model. The sequence length n is variable in a session supported by many RNN training systems.

C. Basic RNN framework

RNN models achieve state-of-art performance in machine translation [14] where a sentence is regarded as a sequence of words. Following this success, the RNN models are extended into many domains that need to exploit the temporal relation in sequence. In this scene, the match sequence indicates the player interaction with the MOBA game system. The player interest is implied in these interactive behaviors. Fig. 1 introduces a basic RNN-based framework to model player interest and only uses three time steps for simplicity. For supporting a long sequence, the LSTM is the main network unit.

LSTM takes a match sequence $\{m_0, m_1, ..., m_{n-1}\}$ as input where every match corresponds to an input state. m_i is a sparse vector by concatenating all event vectors, and an event vector is represented by one-hot encoding, which is similar to many application scenes [6], [21].

The LSTM layer is the non-linear transform of input states. h_t is computed by the current input and the previous state by the following formulation.

$$h_t = LSTM(h_{t-1}, m_t) \tag{12}$$

By this LSTM layer, we obtain the vector representation of sequence s which is the combination of the hidden states.

$$s = f(h_0, ..., h_{n-1}) \tag{13}$$

Google machine translation [14] uses the last hidden state h_{n-1} as the vector representation of word sequence. Here, it represents the high-level feature vector of player interest.

Following the LSTM layer, a Softmax regression model [1] takes h_n as input. The output is a probability vector $O_{out} = \{o_0, o_1, ..., o_{N-1}\}$, where o_i is a real value that represents the probability of purchasing the i-th item.

Although the basic framework completes the extension of RNN in recommendation scenes of a mobile MOBA game, it can not solve some challenges. One challenge is the noise matches that occur in a sequence. In an ideal situation, each match provides positive information for player interest. For *natural language processing* (NLP) applications, the order of words in a sentence is grammatical. However, there are many random player behaviors in a MOBA game. These random behaviors provide negative information for player interest, which requires that the model can degrade the effect of noise behaviors. In addition, the larger challenge is to interpret why recommendations are made. It not only is helpful to guide players to purchase items but also provides player feedback about the current game version for game developers. With these feedback, developers can better understand player interest and design new heroes or skins that are more interesting to players. To solve these challenges, we build an unified framework that can tolerate random behavior and provide reasonable explanation for recommendation results.

D. Hierarchical attention-based LSTM framework

Although LSTM models achieve better prediction performance in many applications, the simple LSTM model lacks the ability to interpret recommendation results. By combination with an attention mechanism, a feasible approach is provided to solve the interpretability problem. An attention mechanism is firstly proposed in machine translation [13] to train the alignment model. When decoding the context, the decoder pays more attention to the most relevant parts of the source sentence. Inspired by this idea, we propose an interpretative

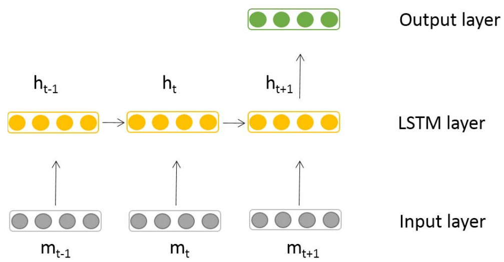

Fig. 1. A basic LSTM-based recommendation framework

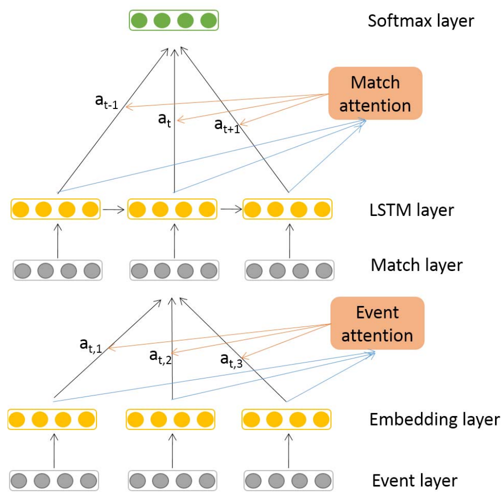

Fig. 2. A hierarchical attention recommendation framework

framework. Unlike an encoding-decoding framework where an input state is a single word, each match in a mobile game includes many player events. The learning framework must not only pay attention to the relevant match but also pay attention to the relevant events. The proposed interpretative framework designs a hierarchical attention mechanism, as shown in Fig. 2. Hierarchical attention refers to match attention and event attention. Match attention evaluates the importance of matches, and the event attention evaluates the importance of events.

1) Event layer: The event refers to the player behavior in a match, such as hero ID, winning state. In many recommendation scene, these events are usually encoded by one-hot encoding and concatenated as a feature vector of user behavior [21]. It must apply the ReLU activation function to enable the learned vector representation to be interpretable. Instead, our proposed framework encodes each event separately and generates the match vector in the attention method.

2) Embedding layer: The attention computes a weighted sum of the event vector, thus requiring the dimension of each vector is the same. However, one-hot encoding of events leads to different vector dimensions. To solve this problem, an embedding layer is designed that converts sparse one-hot vectors to dense vectors with the same dimension. The embedding vector $f_{t,i}$ of event i is computed by the following

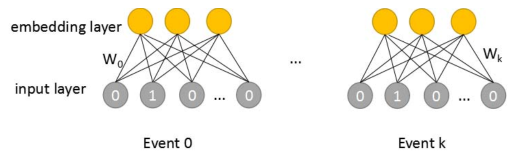

Fig. 3. The embedding architecture

formula.

$$f_{t,i} = w^i e_{t,i} \tag{14}$$

Fig. 3 describes the sub-network structure from the event layer to the embedding layer. w^i is the learned parameters of $e_{t,i}$. The embedding layer is a fully connected architecture, and the total set of parameters $\{w_0, w_1, ..., w_k\}$ is trained with the whole framework. The embedding dimension d must be considered, which affects the recommendation accuracy. In the following experiments, we detail this problem and choose a suitable value of d.

3) Event attention: Event attention is used to represent a match by combining the embedding vectors of events. A match includes many player events and not all events contribute equally to the representation of the match. The idea of attention is to assign different weights to each event. So m_t is represented by the following formulation.

$$m_t = \sum a_{t,i} f_{t,i} \tag{15}$$

where $a_{t,i}$ is the weight of event i in the t-th match. $a_{t,i}$ is computed by

$$a_{t,i} = \frac{exp(u_{t,i}^T u_w)}{\sum_t exp(u_{t,i}^T u_w)} \tag{16}$$

where u_w is a hyper-parameter that is initialized randomly and jointly learned during the training process. $u_{t,i}$ is the hidden representation of $h_{t,i}$ computed by a non-linear transform.

$$u_{t,i} = tanh(W_w f_{t,i} + b_w) \tag{17}$$

where W_w and b_w are learned parameters.

4) Match attention: The LSTM input layer and the LSTM layer are the same as the basic RNN framework. Before a player makes a purchase decision, the player can play many matches. Different matches bring different game experience. Thus, the importance of matches is different for player interest. Match attention is adopted to give more consideration to important matches. The vector representation of s is a combination of all matches.

$$s = \sum a_t m_t \tag{18}$$

where the weight a_t of each match m_t is computed by

$$a_t = \frac{exp(u_t^T u_w)}{\sum_t exp(u_t^T u_w)} \tag{19}$$

where

$$u_t = tanh(W_w m_t + b_w) \tag{20}$$

is a non-linear transform formula of m_t. u_w, W_w, and b_w are learned parameters.

E. Prediction

After match attention, we obtain high-level vector representation s of player interest based on the most recent match sequence. Based on this representation, we build a prediction model.

$$\hat{y} = \sigma(W * s + b) \tag{21}$$

where W and b are the parameters of the output layer, and σ is the Softmax regression model. In this work, we adopt the averaged cross-entropy as the loss function.

$$l = \frac{1}{N}\sum_{i=1}^{n}(-y_i log(p_i)) - (1 - y_i)log(1 - p_i)) \tag{22}$$

where N is the number of training samples. y_i is the label and p_i is the probability of purchasing items.

V. Experiments

A. Data setting

To evaluate the performance of the proposed framework, we take a series of experiments on the actual game data from King of Glory. The logs of a week from June 10th to June 16th, 2017, are collected. The matches played in a day are organized for a purchase session, which is also called a sample. For every match, we extract eleven behaviors, which include hero ID, proficiency level, killing number, dead number, assist number, MVP score, rank, match result, team MVP, enemy team MVP, and type of hero. Finally, we generate approximately 810,000 samples where 730,000 samples for training and 80,000 samples for test. All models are implemented with TensorFlow and trained on a K40.

B. Compared models

We compare the performance of the proposed LSTM model with other methods. Similar to *click through rate* (CTR) prediction [6], we use the *neural network* (NN) with one hidden layer as the baseline model. The last match is the input represented by concatenating the event vector to a long spare vector. To exploit the temporal relation in the player behavior sequence, we build a naive LSTM framework demonstrated in temporal recommendation [12]. To evaluate the attention, *attention-based LSTM* (aLSTM) is designed, which adds an attention layer on the naive LSTM framework. To evaluate the effect of the embedding method, a pair of models *LSTM with embedding* (eLSTM) and *LSTM with embedding and match attention* (eaLSTM) are built. *Hierarchical attention LSTM* (haLSTM) is the proposed framework.

C. Experimental result

1) Top-5 accuracy: The overall performance on the top-5 accuracy is shown in Fig. 4. By default, the LSTM layer has one hidden layer with 500 neurons. The dimension of embedding layer is 64. The early stop strategy is used to avoid over-fitting. When the accuracy on the validation set decreases in five consecutive epochs, the training is stopped and the best accuracy is chosen as the final result. In each iteration, the mini-batch size is 64, and it may take tens of epochs before

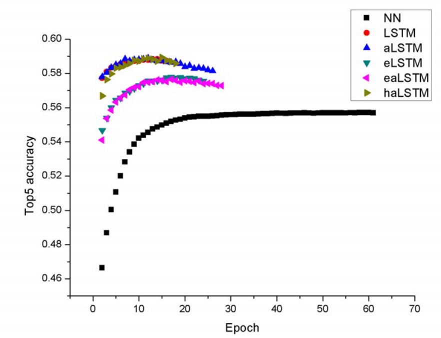

Fig. 4. Top-5 accuracy for different models

the model converges. After tuning, some values are modified to achieve better performance.

From Fig. 4, the NN model obtains the worst performance because it does not consider the historical behavior of players. In addition, it takes more epochs to converge. The LSTM, aLSTM, and haLSTM models achieve the highest accuracy and are 2% better than the NN model. It is reasonable that LSTM models exploit the temporal information of the historical behavior of players. eLSTM and eaLSTM models have lower accuracy than the best model because of the embedding dimension d value, which will be detailed later. Compared LSTM to aLSTM and haLSTM, the attention does not significantly improve the accuracy. To understand the reason for this observation, firstly, we rethink the characteristic of LSTM without attention. In a standard LSTM model the state close to the last state has an important effect. Thus, if the length of the useful history behavior sequence is small, the attention cannot work effectively. This reasoning is consistent with CTR prediction [6] where the length of the user behavior sequence is three. In the next experiment, we demonstrate the effect of sequence length.

2) Effect of the sequence length: The sequence length is an important performance factor, which implies how the historical behavior of players affects future behavior. Long behavior sequences increase the computation overhead of LSTM models, and only a small number of elements in the sequence are useful. Short sequences lose some useful information. To evaluate the effect of sequence length, experiments are conducted by setting the sequence length to 1, 2, 4, 6, and 8. When the length is one, the model only uses the last match as the input. Fig. 5 shows the accuracy with different lengths. The evaluation model is haLSTM that includes match attention and event attention. When the sequence length is two, haLSTM achieves the highest accuracy. As the length is increased to eight, the accuracy is lower in the NN model, which only uses the last match. According to this result, the recent effect plays an important role in predicting future player behavior. The recency effect is a kind of behavior psychology where the recent user behavior has an important effect. In King of Glory, players usually take approximately 20 minutes for a game. Too

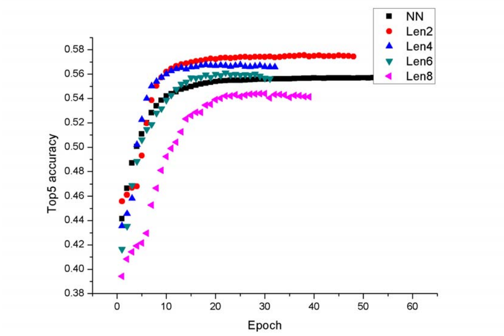

Fig. 5. Top-5 accuracy with fixed length

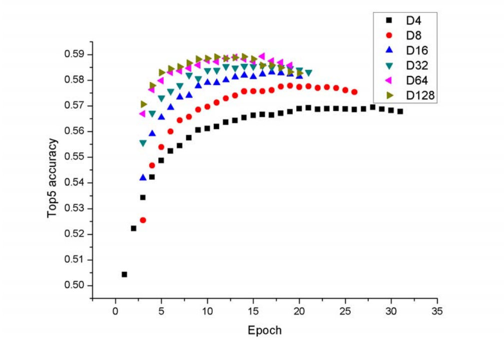

Fig. 6. Top-5 accuracy with variable dimensions

many events occur in a match, and they bring different game experience. The old game experience is easily replaced by the new experience. The new game experience provides useful and precise information for behavior prediction. In a word, the last match has a more important effect in a session.

3) Effect of the embedding method: In the experiment for evaluating overall performance, we find that the LSTM models with an embedding layer have lower accuracy. The effect of embedding dimension d must be considered. Thus, eLSTM are designed by setting d to 4, 8, 16, 32, 64, and 128 in Fig. 6. The result shows that the high dimension embedding vector achieves better accuracy, but higher is not always better. In our scene, the best choice of the dimension is sixty-four. When the dimension increases, the accuracy decreases.

D. Interpretability

The interpretability of the recommendation result is the real requirement for mobile games. When items are rich in scenes, users have difficulty choosing suitable items. The recommender system helps players purchase items by mining the relation of items from user behavior. An interpretive reason makes the recommendation system more persuasive. In mobile games, the players prefer to purchase items that lead to better game experience. The proposed framework solves this problem and reveals the correlation between player behavior and items. The interpretable result tells the players if they purchase these items then they will have a better game experience. By analyzing the statistic of the attention values, we evaluate the interpretability of the proposed framework. The sequence length is six, which is a suitable length because the length of two is too short to analyze. Table I shows the results of match attention and the importance of each match in a sequence. m_i represents the i-th match in a sequence. The number of players is counted with the highest attention value. Obviously, the impact of player historical behavior on future purchase follows the principle of the recency effect. The more recent has a more important effect on future purchase behavior; meanwhile, the player interest is immediate and changes with every match. This proposed framework provides a quantifiable approach for assumptions about player interest.

TABLE I
ATTENTION VALUES OF MATCH ATTENTION.

	m_1	m_2	m_3	m_4	m_5	m_6
player number	7	15	36	121	841	15219

Secondly, we choose eleven player behavior events to evaluate the effectiveness of the event attention in Table II. For all players with correct prediction, the top three event features are counted. The $1st\_playercnt$ represents the player number with the highest values of event attention on the distribution of eleven events. The $2nd\_playercnt$ represents the player distribution with the second highest attention values. The $3rd\_playercnt$ represents the player distribution with the third highest attention values. The hero ID is the first important event for all players. Beside the hero ID, many players are affected by the hero type, game result, and MVP state.

Compared to other events, the hero ID is a strong recommendation feature. The game mechanism of mobile MOBA games is used to explain these results. At the beginning of a game, the player chooses a hero according to his will. Although the players also consider teamwork, the player prefers to choose the hero that he or she likes. This explicit player behavior implies real player interest, which has an important effect. By comparison, other behavior events are the uncertainty factors in a game. A MOBA game is a teamwork game, and the players in a team affect player behavior. For example, the poor team members will lead to game loss and cause a bad game experience. The result also demonstrates the event feature selection is important to achieve better performance. In our eleven events, many events are not useful, such as killing number, dead number. With a hierarchical attention mechanism, the proposed framework can easily train model with different features sets, and then choose the best features. The quantization value of event importance provides a reasonable explanation and helps with feature selection.

VI. CONCLUSIONS

For mobile MOBA games, the purchase preference of players changes over game experience and the interpretability is an actual requirement for mobile game recommender system. To solve these problem, we propose a hierarchical attention-based

TABLE II
ATTENTION VALUES OF EVENT ATTENTION

	hero_id	proficiency	killcnt	deadcnt	assistcnt	mvpscore	rank	result	mvpheorid1	mvpheroid2	herotype
1st_playercnt	16241	0	0	0	0	0	0	0	0	0	0
2nd_playercnt	0	311	0	1	0	3	1096	4239	616	1024	6141
3rd_playercnt	0	2630	0	0	15	53	3013	2956	1631	3046	2897

recurrent neural network recommendation framework. The main skeleton of the framework adopts LSTM to model the long match sequence owing to the special memory structure. The hierarchical attention includes match attention and event attention. Match attention is used to measure the importance of match in a sequence, and event attention is used to measure the importance of events in a match. The experiments demonstrate the effectiveness of attention. In future work, we will combine multiple sessions to model player interest and exploit the effect of historical behavior across sessions of varying player interest.

REFERENCES

[1] Chapelle Olivier, Manavoglu Eren, and Rosales Romer, "Simple and Scalable Response Prediction for Display Advertising", *ACM Transactions on Intelligent Systems and Technology*, 5(4), pp. 1-34, 2013

[2] Posen Huang, Xiaodong He, Jianfeng Gao, Li Deng, and Larry P. Heck, "Learning deep structured semantic models for web search using clickthrough data", *in Proceedings of the 22nd ACM International Conference on Conference on Information and Knowledge Management (CIKM)*, pp. 2333-2338, 2016

[3] Sarwar Badrul, Karypis George, Konstan Joseph, and Riedl John, "Item-based collaborative filtering recommendation algorithms", *in Proceedings of the 10th International Conference on World Wide Web (WWW)*, pp. 285-295, 2001

[4] Jiahui Liu, Peter Dolan, and Elin Rnby Pedersen, "Personalized news recommendation based on click behavior", *in Proceedings of the 15th International Conference on Intelligent User Interfaces*, pp. 31-40, 2010

[5] Paul Covington, Jay Adams, and Emre Sargin, "Deep Neural Networks for YouTube Recommendations", *in Proceedings of the 10th ACM Conference on Recommender Systems*, pp. 191-198, 2016

[6] Yuyu Zhang, Hanjun Dai, Chang Xu, Jun Feng, Taifeng Wang, Jiang Bian, Bin Wang, and Tieyan Liu, "Sequential click prediction for sponsored search with recurrent neural networks", *in Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pp. 1369-1375, 2014

[7] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F. Stewart, and Jimeng Sun, "Doctor AI: Predicting Clinical Events via Recurrent Neural Networks", *in Proceedings of the 1st Machine Learning in Health Care*, pp. 301-318, 2016

[8] Cristobal Esteban, Oliver Staeck, Stephan Baier, Yinchong Yang, and Volker Tresp, "Predicting Clinical Events by Combining Static and Dynamic Information Using Recurrent Neural Networks", *in Proceedings of the 2016 IEEE International Conference on Healthcare Informatics (ICHI)*, pp. 93-101, 2016

[9] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk, "Session-based Recommendations with Recurrent Neural Networks", *in Proceedings of the 6th International Conference on Learning Representations*, pp. 100-108, 2015

[10] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk, "Parallel Recurrent Neural Network Architectures for Feature-rich Session-based Recommendations", *in Proceedings of the 10th ACM Conference on Recommender Systems (RecSys)*, pp. 241-248, 2016

[11] Sai Wu, Weichao Ren, Chengchao Yu, Gang Chen, Dongxiang Zhang, and Jingbo Zhu, "Personal recommendation using deep recurrent neural networks in NetEase", *in Proceedings of the 32nd IEEE International Conference on Data Engineering (ICDE)*, pp. 1218-1229, 2016

[12] Yang Song, Ali Mamdouh Elkahky, and Xiaodong He, "Multi-Rate Deep Learning for Temporal Recommendation", *in Proceedings of the 39th International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 909-912, 2016

[13] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate", *in Proceedings of the 3rd International Conforence on Learning Representations (ICLR)*, pp. 91-100, 2015

[14] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le, "Sequence to Sequence Learning with Neural Networks", *in Proceedings of the 27th Annual Conference on Neural Information Processing Systems*, pp. 3104-3112, 2014

[15] Alex Graves, Abdel Rahman Mohamed, and Geoffrey E. Hinton, "Speech recognition with deep recurrent neural networks", *in Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6645-6649, 2013

[16] Tengteng Guo, Lidan Wang, Menghe Zhou, and Shukai Duan, "A recurrent neural network based on memristive activation function and its associative memory", *Science China Information Sciences*, 47(9), pp. 1226-1241, 2017

[17] Jinde Cao, and Ruoxia Li, "Fixed-time synchronization of delayed memristor-based recurrent neural networks", *Science China Information Sciences*, 60(3), pp. 1110-1125, 2017

[18] Thang Luong, Hieu Pham, and Christopher D. Manning, "Effective Approaches to Attention-based Neural Machine Translation", *in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1412–1421, 2015

[19] Shuangfei Zhai, Kenghao Chang, Ruofei Zhang, and Zhongfei Zhang, "DeepIntent: Learning Attentions for Online Advertising with Recurrent Neural Networks", *in Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining*, pp. 1295-1304, 2016

[20] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter F. Stewart, "RETAIN: An Interpretable Predictive Model for Healthcare using Reverse Time Attention Mechanism", *in Proceedings of the 29th Annual Conference on Neural information Processing Systems*, pp. 3504-3512, 2016

[21] Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao, "Dipole: Diagnosis Prediction in Healthcare via Attention-based Bidirectional Recurrent Neural Networks", *in Proceedings of the 23rd International Conforence on Knowledge Discovery and Data Mining*, pp. 1903-1911, 2017