



# Article Sequential Recommendations on Board-Game Platforms

# JaeWon Kim, JeongA Wi, SooJin Jang and YoungBin Kim \*D

Department of Image Science and Art, Chung-Ang University, Dongjak, Seoul 06974, Korea; jaewon.k92@gmail.com (J.K.); placeja@cau.ac.kr (J.W.); sujin0110@cau.ac.kr (S.J.)

\* Correspondence: ybkim85@cau.ac.kr

Received: 25 December 2019; Accepted: 23 January 2020; Published: 2 February 2020



**Abstract:** The game market is an increasingly large industry. The board-game market, which is the most traditional in the game market, continues to show a steady growth. It is very important for both publishers and players to predict the propensity of users in this huge market and to recommend new games. Despite its importance, no study has been performed on board-game recommendation systems. In this study, we propose a method to build a deep-learning-based recommendation system using large-scale user data of an online community related to board games. Our study showed that new games can be effectively recommended for board-game users based on user big data accumulated for a long time. This is the first study to propose a personalized recommendation system for users in the board-game market and to introduce a provision of new large datasets for board-game users. The proposed dataset shares symmetric characteristics with other datasets and has shown its ability to be applied to various recommendation systems through experiments. Therefore, the dataset and recommendation system proposed in this study are expected to be applied for various studies in the field.

**Keywords:** board game; dataset; deep neural network; implicit feedback; recommendation system; sequential recommendation systems

# 1. Introduction

The game market is expanding because of advances in technology, platforms, and devices. The board-game market has naturally gained popularity at an exponential rate, as shown in Figure 1. However, including the board-game market, the system of recommendation of the game market is a relatively new area [1]. Even a well-known game distribution platform such as Steam has just recently adopted AI user-based recommendations to allow game communities to search for a new game in less time and by spending less effort along with user acquisition cost for game developers [2,3]. The board-game market is rapidly growing because of the rising demand for traditional games (board games) in the game market [4,5]. However, recommendation systems have not been actively studied in this field.

Interactions such as purchasing and playing games are more like listening to music than watching a movie [6] because every game purchase of users and interplays are recurrent actions. Specifically, users seek to play their favorite games many times but also hope to discover new games at the same time (like listening to music). Generally, for choosing a new game, a customer is impelled to skim through game reviews or ratings to decide whether he or she will like the game unless he actually plays the game. Therefore, an effective recommendation system can alleviate the user's inconvenience and save user acquisition cost.



Figure 1. Rising popularity in the board-game market (https://www.kickstarter.com/).

Among the various board-game communities, BoardGameGeeks (BGG) is one of the largest online forums for board gaming. It is composed of an abundant game database, which contains over 100,000 board games and 2 million registered users as of February 2019. In addition, the number of reviews has reached almost 50 million. In these reviews, the players share information such as their impression of the games based on their acquisition status (e.g., for trade, own, preowned, want, want to buy, wish list). We present a large database using the above information from the BGG board-game community. Although there is a vast amount of data available on the BGG, our dataset (BGG dataset) employed the status information from the reviews to capture the user's interests and historical sequence representations. Accordingly, various approaches have been proposed for recommendation systems to propose a list of games that matches the user's interest in the board-game industry.

A recommendation system is widely used in various industries, for instance, e-commerce, online advertising, social media, and media streaming services. Recommender systems are algorithms that suggest the relevant or favored items to users. As recommendation systems are used in various fields and the amount of content dramatically increases, the effective recommendations are significantly essential. Subsequently, considerable research has actively been conducted on this topic. The introduction of deep learning-based recommendations has considerably contributed to the prominence of the studies. Various models and techniques such as neural collaborative filtering (NCF), neural factorization machine (NFM), recurrent neural network (RNN), convolutional neural network (CNN), and reinforced learning model have been proposed. Each model recommends items based on different points of attention concerning their specific tasks. Some systems address user-based settings, whereas others address item-based settings regarding users' general preferences, personalization, or sequential patterns. The recent recommendation systems such as convolutional sequence embedding recommendation model (Caser) [7] and self-attentive sequential recommendation (SASRec) [8] use the interaction sequences that contain fruitful information about each user's behavior [9], e.g., music listening, purchasing merchandise [10], watching YouTube [11], and playing video games [2], and the similarity between the items by capturing both long-range and short-range dependencies of user-item interaction sequences.

In this study, we fed recommendation algorithms with the BGG dataset (https://github.com/ John-K92/Recommendation-Systems-for-BoardGame-Platforms) to test whether recommendations can properly obtain sequential interactions in the board-game market from a large-scale dataset. We implemented a deep neural network(DNN)-based sequential recommendation algorithm of gated recurrent units for recommendation (GRU4Rec) [12], Caser, and SASRec [8] with their variations to evaluate our dataset. Based on sequential recommendation studies [7,8,13,14], we hypothesize that solely feeding the models with item preference sequences of users is redundant to test recommendations in the board-game market. In summary, we achieved precision scores of 0.19 and 0.06 and NDCG scores of 0.36 and 0.14 for 10 items when cold starters are set to 200 and 8 interactions, respectively. The contributions of the study are as follows:

- To the best of our knowledge, we are the first to propose recommendation systems for board-game platforms.
- We present a large-scale dataset of user-game interactions from BGG for research in games. It contains 108,536 board games and 511,960 users with over 47 million ratings.
- We provide the potential of sequential recommendations in the board-game platform.

## 2. Related Work

In this section, we discuss related works from three aspects: recommendations in game markets, general recommendation systems, and sequential recommendation systems.

## 2.1. Recommendations in Game Markets

Numerous studies have greatly facilitated our daily life; however, just a few have focused on the game industry. The game communities, e.g., Steam (https://store.steampowered.com/), GOG (https://www.gog.com/) (Good Old Games), and BGG, are appealing platforms to develop and apply recommendation systems. Most pioneering game recommenders were based on traditional machine learning algorithms. Among them, extremely randomized trees (ERTs) [2], which are a randomized version of ensemble trees, have been pivotal. Because prior to the introduction of graphics processing units computing power, ERTs were computationally efficient with a high degree of parallelization; moreover, they are robust against over-fitting in tasks such as recommendations. Furthermore, NLP-based algorithms such as those using text-reviews from users [15,16] have been widely implemented.

As in other applications, the trend of the recommendation systems moved to the implementation of DNN-based algorithms. Quadrana et al. [17] introduced a recommendation system that highlighted the potential of neural networks in the video game industry. Subsequently, Wan et al. [18] studie monotonical behavior chain concept that a more explicit preference toward items implies more implicit and abundant representations. This work shifted the paradigm that a recommendation task can have potential over various datasets and fields including the game community area. Therefore, recently, the case study on Steam platform [2] compared state-of-the-art techniques from implicit feedback using collaborative filtering (CF) [19], factorization machine (FM) [20], and DNN-based variational FM models [18]. These previous studies inspired us to implement recommendations in the board-game market where no research has been conducted yet.

#### 2.2. General Recommendation Systems

Generally speaking, recommendation systems use CF to recommend unseen items that other users with a similar preference liked based on interaction histories [21–23]. Although the various algorithms have been substituted by modern algorithms such as deep learning-based models because of their performance, non-negative matrix factorization is a prevalent method that predicts the user's interest on items by projecting items and users into a latent space by the inner product of the vectors via exploitation of global information [22]. Another popular algorithm in CF is two-layer Restricted Boltzmann Machines [24], which are a pioneering method that uses DNNs and a winner of the Netflix Prize (https://www.netflixprize.com). Along the lines of DNN, neural collaborative filtering [25], AutoRec [26], and CDAE [27] using auto-encoder have replaced conventional algorithms. However, the reason for the decline of these systems is their inability to represent sequential information.

#### 2.3. Sequential Recommendation Systems

In the game market, to be specific, the integration of recommendation systems is a relatively new area. Despite every game purchase of users and playing activities being a recurrent action [6],

the suggestions are solely made based on their played games [1]. Thus, the use of sequential interactions in games is essential information for recommendation. The sequential recommendation systems aim to predict the successive items by the users' historical sequential interaction patterns in time order. The rising popularity of such systems is because of its robustness to the cold-start problem and not ignorable information within the user's sequential actions (e.g., item purchases, music listening, and streaming videos). Early approaches are the Markov chain-based models [28–30], where recommendations are made based on L previous actions with an L-order Markov chain. The paradigm then moved to RNN-based models such as GRU4Rec [12] or recurrent recommender network(RRN) models [31] to capture representations in user behavior sequences. These approaches have undoubtedly received attention because RNN shows impressive capabilities when modeling sequential data [32]. In contrast, CNN-based models have been introduced as an option to learn sequential patterns, such as Caser proposed by Tang and Wang [7] using both horizontal and vertical convolution filters to address RNN's vulnerability towards the gradient vanishing problem while modeling sequences.

Recently, some studies attempted to employ the neural attention mechanism to represent sequential interactions and improve recommendation performance [33,34]. The attention mechanism learns to pay attention only to the important features in a given input; thus, it has been considerably studied in natural language processing (NLP) and computer vision. Furthermore, the self-attention mechanism [35], unlike standard vanilla attention, models the complex sentence structure and retrieves relevant words in generating the next word by considering their relations. Several approaches for using the self-attention mechanism [8,14,36] have been proposed to represent users' sequential behavior through a high-level semantic combination of elements and have achieved state-of-the-art results.

#### 3. Dataset and Models

#### 3.1. BGG Dataset

Prior to the implementation of the recommendation systems, the dataset obtained from the BGG database will be discussed first. The database comprises board games with their metadata such as title, ratings, production year, publisher, types, categories, and other details. From the above database, we constructed the dataset as in Tables 1 and 2. While creating the dataset, we assumed that the user interacts with the game (add a rating or comment) when he or she purchases, plays, or at least makes a review in sequence. This conjecture was made because the number of play logs of the users was missing. They may upload the logs online, but because there was no synchronization with the real play logs, the users must upload at their own will. Additionally, BGG is not a retail or publishing platform of the board games, so we could not retrieve the exact game purchase date and time. Therefore, playtime or purchase information is not a very accessible feature from the BGG database.

Various features need to be expanded more. First, the genre of each game is divided into two features, "type" and "category." Type comprises eight unique values, whereas category contains 84 values. Thus, type can be perceived as being more comprehensive than feature. It will be elaborated more in the next section. In addition, both type and category features are not a unique key feature, i.e., they may not have any value in the list or may have multiple values in the list. Second, the "status" feature in the user rating dictionary may also have multiple values, for example, it can have values such as for trade and own status. Lastly, ratings are composed of user ratings and user comments at the same time, but no duplicate users in the ratings. This means that one user may rate the game and create a comment separately, but only the latest one with a rating score will be visible on the list with its status.

BGG Dataset								
Feature	Туре	Description						
game_id	Int	Unique id of the game						
game_title	Str	Title of the game						
year	Int	Production year						
people	Str	A range of the number of people recommended playing with						
play_time Int		The approximate time duration for a gameplay						
age Str		Information on age restriction						
BGG_ratings	[Str]	Average ratings from users						
designer	[Str]	Designers of the game						
artist	[Str]	Visual artists of the game						
publisher	[Str]	Publisher of the game						
type	[Str]	Types of the game						
category	[Str]	Categories of the game						
User ratings	Dict	User rating dictionary						

Table 1. Summary of BGG Dataset features.

Table 2. Summary of features in User ratings.

User Ratings								
Feature	Туре	Description						
user	Str	A key for unique usernames						
score	Int	A score for the ratings						
status	[Str]	Status of acquisition of the game						
time_stamp	smalldatetime	Rating/comment created time						

Other well-known datasets such as Steam, MovieLens, and Amazon (http://jmcauley.ucsd. edu/data/amazon/) with respect to recommendation tasks are briefly compared using the statistics of each dataset in Table 3. For all datasets, the constructed interactions have been assigned from a review or rating (implicit feedback) of each user and item. The domains and sparsity of each real-world representative dataset vary significantly. However, it can easily be noted that the BGG dataset is the largest among the described ones. Considering its size, relative sparsity, and average interactions, the BGG dataset would be an interesting database to research and develop board-game recommendation systems.

Table 3. Statistics of datasets.

Datasets	#users	#items	#interactions	avg.inter./user	avg.inter./item	Sparsity
ML 1M	6022	3043	999,154	165.25	327.03	94.57%
ML 20M	138,493	18,345	19,984,024	144.30	1089.34	99.21%
Beauty	22,363	12,101	198,502	8.88	16.40	99.93%
Video Games	108,063	31,390	21,105,584	9.54	21.72	99.38%
Steam	62,944	9198	5,132,610	81.54	558.01	99.11%
BGG	388,413	87,208	47,351,708	121.91	542.97	99.86%

## 3.2. Recommendation Models

In this study, the sequential recommendation task assumes a set of users  $\mathcal{U} = \{u_1, u_2, ..., u_{|\mathcal{U}|}\}$  and a set of games  $\mathcal{G} = \{g_1, g_2, ..., g_{|\mathcal{G}|}\}$ . The interaction sequence in chronological order of each user is denoted as  $S^u = [g_1^u, ..., g_t^u, ..., g_{|\mathcal{S}^u|}]$ , where  $g_t^u \in \mathcal{G}$  is the game that user u interacted with, at a time step t. Similar to Caser, BERT4Rec [14], and other sequential-based methods [37,38], t denotes the order of the interactions rather than the absolute timestamp in temporal recommendation systems [31,39,40].

## 3.2.1. GRU4Rec

As a variant of RNN, gated recurrent units (GRUs) are used for modeling interaction sequences in a session-based scenario. GRU [41] is a more elaborate version of RNN that can solve the vanishing gradient problem of RNN with an update gate and a reset gate. The GRU gate basically learns the amount of information to update and forget the hidden state of the unit throughout each gate. The update gate is calculated using x as an input plugged into the network unit, given that **W** and **U** represent the learnt weight at each gate, with h being the information of the hidden gate at a given time step t:

$$z_t = \sigma(\mathbf{W}_z x_t + \mathbf{U}_z h_{t-1}) \tag{1}$$

where  $\sigma(\cdot)$  is a sigmoid function for non-linear transformation. The reset gate is given by:

$$r_t = \sigma(\mathbf{W}_r x_t + \mathbf{U}_r h_{t-1}). \tag{2}$$

A memory content that uses the reset gate to store the information from the previous one is given by:

$$\hat{h}_t = tanh(\mathbf{W}x_t + r_t \odot \mathbf{U}h_{t-1}) \tag{3}$$

where  $\odot$  is an element-wise product of matrices. The final linear interpolation between the past and current step to determine what to store from the current memory content is given as follows:

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \hat{h}_t.$$
(4)

This GRU-based recommendation system produces one of the highest performance in RNN-based algorithms. The brief architecture of GRU4Rec is in Figure 2. In the figure, *L* and *T* represent the previous and target item sequences, respectively, within a user interaction sequence  $S^u$ . **E** denotes the embedding matrix of *L* previous sequences. Hidashi et al. [37] introduced new loss functions, TOP-1 loss and BPR, and mini-batch negative sampling to capture the inter-dependencies to further improve the recommendation accuracy.



Figure 2. Model architecture of GRU4Rec.

# 3.2.2. Caser

The model leverages both horizontal and vertical convolutional filters to learn sequential representations from interactions. It uses a latent factor for the CNN. Tang and Wang [7] proposed convolution processing to treat an embedding matrix  $\mathbf{E} \in \mathbb{R}^{\mathcal{L} \times d}$  as the image, where  $\mathcal{L}$  denotes the previous interacted items in *d*-dimensional latent space. Thus, the model can be fed with local features of the image as a sequential pattern and can additionally learn general preferences. The horizontal convolution is calculated as  $\phi_c(\mathbf{E}_{i:i+h-1} \odot \mathbf{F}^k)$  where  $\phi_c$  denotes the activation function

for convolutional layers and *h* is the height of a filter, referring to the illustrated model architecture in Figure 3. The vertical convolution is attained from  $\sum_{l=1}^{L_{l=1}} \tilde{\mathbf{F}}_{l}^{k} \cdot \mathbf{E}_{l}$ . Unlike the horizontal convolutional layer, each filter size is fixed at  $1 \times L$ . Then, the outputs of the two convolutional layers are fed into a fully connected (FC) layer to restore more high-level and abstract features. The output is then concatenated with the user's general preference and the  $\mathbf{P}^{u}$  embedding, whose result is then projected to an output layer.

$$\mathbf{y}^{(u,t)} = \mathbf{W}' \begin{bmatrix} z \\ \mathbf{P}^u \end{bmatrix} + b',\tag{5}$$

where W' and b' are the weight matrix for the output layer and the bias term, respectively.



Figure 3. Model architecture of Caser.

### 3.2.3. SASRec

Kang and McAuley [8] used self-attention networks with single head attention of a left-to-right unidirectional model; the model architecture can be found in Figure 4. It identifies the items that are relevant or important from a sequential history and models to predict the next time in sequence. Compared with the above CNN- and RNN-based models, SASRec is cost-efficient and can capture syntactic and semantic patterns between items. The model has a general tendency to focus on long-range dependencies in a high-density dataset; however, it simultaneously focuses on recent activities in sparse datasets. Unlike the Caser model, the item embedding matrix in SASRec is aware of the positions of previous items by positional encoding of each interaction item. The inputs in the self-attention block are query (**Q**), keys (**K**), and values (**V**). The attention of the scaled dot-product is calculated as

Attention(
$$\mathbf{Q}, \mathbf{K}, \mathbf{V}$$
) = Softmax( $\frac{\mathbf{Q}\mathbf{K}^{\mathsf{T}}}{\sqrt{d}}$ ) $\mathbf{V}$ . (6)

For the output of the weighted sum of all values of given interactions between query and key, the scale factor of  $\sqrt{d}$  is used to prevent the large values of the inner product, where *d* is the-dimensionality. Because the queries, keys, and values in self-attention use the same objects, SASRec takes the embedding **E** as an input. The equation for each self-attention head is given as

$$\mathbf{A} = Attention(\mathbf{EW}^{Q}, \mathbf{EW}^{K}, \mathbf{EW}^{V}).$$
(7)

The output is then applied with a point-wise two-layer feed-forward network sharing the parameters. Thus, a self-attention block can be written as:

$$\mathbf{F}_i = FFN(\mathbf{A}_i) = ReLU(\mathbf{A}_i\mathbf{W}^1 + b^1)\mathbf{W}^2 + b^2.$$
(8)

Thus,  $F_i$  aggregates all sequential history of items. However, Kang and McAuley additionally argued that the model can learn more complex item sequential information via stacking another self-attention block.



Figure 4. Model architecture of SASRec.

## 4. Experiments

In this section, we first introduce a preprocessed BGG dataset along with its statistics and evaluate the recommendation systems and their variations. Then, we elucidate our report and analyze the experimental results. For a fair comparison, we imported the code provided by each corresponding author.

# 4.1. Experimental Setup

# 4.1.1. Dataset

We only focus on the interaction between a user and a game. Therefore, special attention is given to interactions with "own" and "preowned" labels in the game acquisition status under user ratings dictionary by converting all numeric ratings to implicit feedback of 1 while separating the other labels. As mentioned in the above section (Dataset and Methods), we also removed the cold-start users and items with less than *n* interactions. This way of overcoming the cold-start issue is a common practice in recommendation systems as in [7,25,29,38,42]. *n* is set to 8 and 200. The main reason for having *n* as 8 is that we maintained the minimum length of *L* previous interactions as 5 and a length of 3 for target sequences *T*. Thus, when the value of *n* is 5, it is not ideal for our model because there are no target sequences left for training. As a result, a summary of statistics of the final two datasets for the training is presented in Table 4. The number of reviews for each dataset is changed to 33,564,515 and 16,090,930 from the total of 47,623,627 reviews, with respect to *n* values. A large sparsity is not an ideal concept for effective training; therefore, we set *n* to two types to reduce the sparsity. Although the original size has decreased to around 16 million from 47 million, it is still a considerable amount to evaluate the recommendations. From the given data, 20% and 10% of each *n*-length dataset was split for testing and validation, respectively.

 Table 4. Summary of statistics of preprocessed datasets.

Descriptive Statistics									
min. number of interactions	8	200							
number of interactions	33,564,515	16,090,930							
number of unique users	322,061	36,164							
number of unique games	59,746	14,686							
data sparsity	99.83%	96.97%							
average interactions per user	104.22	444.94							
average interactions per game	561.76	1095.66							

When it comes to applying recommendation systems in board-game platforms with the BGG dataset, special attention needs to be paid to setting the minimum number of interactions and modeling additional information such as categories and types. First, similar to other recommendation datasets, the BGG dataset is biased with only a few games that account for a large amount of interactions. Figure 5a shows that around 25% of games explain almost 90% of the interactions between users and games. In contrast, the user side is relatively equally distributed, referring to Figure 5b, which the degree of information loss in setting the *n* value is precisely different. Next, the use of labeled types of games is not recommended. As previously mentioned, types and categories of the games are multi-labeled features. However, "None" value is not included in this assertion. Figure 6 illustrates the number of games under each type. Notably, most games are under the None value, meaning that the recommendation system may be inclined to predict and recommend most games with None type items. In contrast, Figure 7 shows that the categories feature has a less tendency to fall into the biased taxonomy. Therefore, heuristically and statistically, for those who are interested in using the BGG dataset, we suggest using either only the categories feature or both.



(a) Cumulative sum of game interactions(b) Cumulative sum of user interactionsFigure 5. Statistics of the number of games and the number of users.



Figure 6. Number of games for each "type" feature.



Figure 7. Histogram plot of the "categories" feature.

#### 4.1.2. Evaluation Metrics

We evaluated the recommendation systems by various evaluation metrics, including precision@*N*, recall@*N*, mean average precision (MAP), normalized discounted cumulative gain (NDCG@*N*), and mean reciprocal rank (MRR) similar to the studies by in [7,8,14,28,43,44]. Precision@*N* and Recall@*N* are computed by

$$Prec@N = \frac{|\mathcal{R} \cap \mathcal{R}_{1:N}|}{N},$$

$$Recall@N = \frac{|\mathcal{R} \cap \mathcal{R}_{1:N}|}{|\mathcal{R}|},$$
(9)

where  $\hat{\mathcal{R}}_{1:N}$  denotes the top-*N* predicted items for a user. MAP is the mean of the average precision (AP) of all users in  $\mathcal{U}$ , given rel(N) = 1 if the predicted *N*-th item in  $\hat{\mathcal{R}}$  is the ground truth  $\mathcal{R}$  when AP is assigned by

$$AP = \frac{\sum_{N=1}^{|\hat{\mathcal{R}}|} \operatorname{Prec}@N * \operatorname{rel}(N)}{|\hat{\mathcal{R}}|}.$$
(10)

NDCG@*N* is a position-aware metric, i.e., measuring a ranking quality by assigning weights in regards with the ranks. In recommendation tasks, NDCG@*N* is calculated as

$$NDCG@N = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{2_{\mathcal{R}_{u,g_u}}^{rel} - 1}{\log_2(\mathcal{R}_{u,g_u} + 1)},$$
(11)

where  $g_u$  indicates the ground truth item for a user and  $\mathcal{R}_{u,g_u}$  is the predicted rank for  $g_u$  given user u. MRR explains how well the recommendation system assigns a rank similar to NDCG. To differentiate two metrics, MRR focuses on the evaluation of assigning the ground truth item a higher rank in prediction, but NDCG considers the rank in relation to adding larger weights on higher ranks. MRR is defined as follows

$$MRR = \frac{1}{M} \sum_{u \in \mathcal{U}} \frac{1}{\mathcal{R}_{u,g_u}}.$$
(12)

Models	_			n = 200						
	Prec@10	Recall@10	MAP	MRR	NDCG@10	Prec@10	Recall@10	MAP	MRR	NDCG@10
GRU4Rec	0.003	0.005	0.003	0.002	0.011	0.082	0.022	0.037	0.011	0.1873
Caser	0.061	0.112	0.074	0.053	0.142	0.1921	0.052	0.092	0.025	0.355
SASRec	0.058	0.108	0.07	0.05	0.135	0.175	0.047	0.079	0.023	0.344

Table 5. Recommendation performance comparison of different models on BGG dataset.

#### 4.2. Performance Analysis

The results of the previously discussed recommendation algorithms on the BGG dataset are given in Table 5. The CNN-based model outperformed all models in every matrix. In general, the self-attention-based recommendation algorithms such as SASRec, BERT4Rec, and AttRec [36] predict next items better than the CNN-based recommendation systems, especially in sparse datasets because CNN-based systems have a limited receptive field [8]. This result reveals that the existence of the relatively weighted short-term sequential interactions in the board-game market. However, the RNN-based GRU4Rec model performed worse than the CNN-based or the self-attention-based algorithms. The primary cause for this result is that the model learns the sequences in a step-by-step method, which means that too much focus on the sequences leads to lower performance. Thus, it needs to capture the dependencies in the short term effectively as well as in the long term (e.g., users' general preference) [7,8,13,38] information. Dealing with the different number of cold-start interactions n, Caser surpassed in Prec@10 and MAP in 200 interactions and above. When n = 8, these scores were very close to each other, but the gap was enlarged because we concentrated more on users with a larger number of interactions. This clearly implies that CNN-based Caser captures the users' short-term sequential preferences while grasping on the general preferences more accurately even with a less receptive field.

Figure 8 shows an example of the predicted games from randomly sampled users in the validation dataset. We fed five sequentially interacted games as the input and presented the highest ten predicted games based on the high-ranking probability. As we predict the game at a time step t, it can be seen that not only the item at the previous time step t - 1 affects the predicted item but also the various time steps in DNN-based recommender. For example, the highest ranked  $item_{t,r}$  does not contain any features ("type" or "category") of an immediately prior item; rather, it is closer to the second and the third sequentially previous items. In addition, the second highest ranked item is particularly close to  $item_{t-4}$ . This shows that results contain various aspects in terms of sequences and the model can learn the sequential and the general preference of a user. Figure 8 indicates the concept of DNN-based sequential recommendation as well as the correlation of class features ("type" and "category") in a board-game recommendation task.

Overall, it can be concluded that an online board-game platform has a unique distinction among various applications in recommendation tasks. Furthermore, it needs to be further investigated to obtain an efficient recommendation system.

item<sub>t,r-9</sub>

Family,

Strateg

Fantasy, Fighting, Territory

Building

Wargame

					•								
Input games sequence it			ames nce <i>ite</i>	item <sub>t-5</sub> item <sub>t-</sub>		item <sub>t-3</sub> i		item <sub>t-2</sub>		iten	tem <sub>t-1</sub>		
	"type"			ategy	gy Thematic, Party Strategy, Thematic		Strat	tegy					
		"catego	An Card City E Civil	cient, Game, Building, ization	Fantasy, Miniatures, Wargame	Bluf Dedu Murde ei Party	fing, ction, r/Myst ry, Game	Adver Fant Figh Warg	nture, asy, ting, jame	Dio S.	ce, F.		
				Re	comme Syste	ndat em	tion						
				Р		l Item	าร						
d ank	item <sub>t,r</sub>	$item_{t,r-1}$	$item_{t,r-2}$	item <sub>t,i</sub>	<sub>r-3</sub> item	t,r-4	item <sub>t</sub>	,r-5	item <sub>t</sub>	;,r-6	item <sub>t</sub>	,r−7	item <sub>t,r-8</sub>
	Thematic	Family, Strategy	Strategy	Strateo Thema	gy, itic Strat	tegy	Strate	egy	Them	atic	Fam	ily	Strategy
y"	Bluffing, Deduction, Horror, Zombies	Ancient, Card Game, City Building, Civilization	Fantasy, Fighting, Miniature, Mythology	Bluffin Negotiat S.F., Space Explorat	ig, Anci tion, Figh e Minia tion Myth	ient, ting, ature, ology	Ancie Card G City Bui Civiliza	ent, ame, Iding, ition	Adven Explora Fanta Fight Minia	ture, ation, asy, ing, ture	Dice Fighti Movie S.F	e, ng, s/TV,	Bluffing, Fighting, Miniature, S.F., Wargame

Input items

Figure 8. An example of predicted items.

## 5. Conclusions and Future Work

Predicte

"type

"category

games in I

In this study, we present a deep learning-based recommendation system using large-scale user data related to the board game. The rampant growth rate of the board-game platform and the distinct aspects imply that more research can be performed in this specific area. Therefore, we explored the BGG dataset with the DNN-based recommendation systems. The mentioned recommenders were evaluated within the context of recommendations in a board-game platform with the past user-game interaction sessions. The results suggest that regardless of the size of the cold-start interactions, CNN-based approach outmatches other recommendation systems and captures personalized preferences in the board-game platform.

Consequently, our future work includes studying various additional state-of-the-art recommendation models to further analyze the attributes in the board-game interactions and the use of multi-labeled features (e.g., game types and categories), which can be an essential key in analyzing the user's interactions and predicting future actions.

Author Contributions: Conceptualization, Y.K.; methodology, J.K. and J.W.; software, J.K.; validation, J.W.; formal analysis, J.W.; resources, J.K.; writing-original draft preparation, J.K., S.J. and Y.K.; writing-review and editing, Y.K.; visualization, J.K. and S.J.; supervision, Y.K.; project administration, Y.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Chung-Ang University Excellent Student Scholarship and the Chung-Ang University Research Scholarship Grants in 2018.

Conflicts of Interest: The authors declare no conflict of interest.

## References

- 1. Sifa, R.; Drachen, A.; Bauckhage, C. Large-scale cross-game player behavior analysis on steam. In Proceedings of the Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference, Santa Cruz, CA, USA, 14-18 November 2015.
- Cheuque, G.; Guzmán, I.; Parra, D. Recommender Systems for Online Video Game Platforms: The Case of 2. STEAM. In Proceedings of the 2019 World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 763-771.

- Anwar, S.M.; Shahzad, T.; Sattar, Z.; Khan, R.; Majid, M. A game recommender system using collaborative filtering (GAMBIT). In Proceedings of the 2017 14th International Bhurban Conference on Applied Sciences and Technology (IBCAST), Azad Rashmir, Pakistan, 10–14 January 2017; pp. 328–332.
- 4. Arizton. Available online: https://www.reportlinker.com/p05482343/Board-Games-Market-Global-Outlook-and-Forecast.html (accessed on 24 September 2019).
- 5. Roeder, O. Crowdfunding Is Driving A \$196 Million Board Game Renaissance. Available online: https://fivethirtyeight.com/features/crowdfunding-is-driving-a-196-million-board-game-renaissance (accessed on 20 September 2019).
- Hariri, N.; Mobasher, B.; Burke, R. Context-aware music recommendation based on latenttopic sequential patterns. In Proceedings of the Sixth ACM Conference on Recommender Systems, Dublin, Ireland, 9–13 September 2012; pp. 131–138.
- Tang, J.; Wang, K. Personalized top-n sequential recommendation via convolutional sequence embedding. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, Los Angeles, CA, USA, 5–9 February 2018; pp. 565–573.
- 8. Kang, W.C.; McAuley, J. Self-attentive sequential recommendation. In Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM), Singapore, 17–20 November 2018; pp. 197–206.
- 9. Cheng, Z.; Shen, J.; Zhu, L.; Kankanhalli, M.S.; Nie, L. Exploiting Music Play Sequence for Music Recommendation. *IJCAI* 2017, *17*, 3654–3660.
- 10. Smith, B.; Linden, G. Two decades of recommender systems at Amazon. com. *IEEE Internet Comput.* **2017**, 21, 12–18. [CrossRef]
- 11. Covington, P.; Adams, J.; Sargin, E. Deep neural networks for youtube recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016; pp. 191–198.
- 12. Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; Tikk, D. Session-based recommendations with recurrent neural networks. *arXiv* 2015, arXiv:1511.06939.
- Yuan, F.; Karatzoglou, A.; Arapakis, I.; Jose, J.M.; He, X. A Simple Convolutional Generative Network for Next Item Recommendation. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, Melbourne, Australia, 11–15 February 2019; pp. 582–590.
- 14. Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; Jiang, P. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. *arXiv* **2019**, arXiv:1904.06690.
- 15. Meidl, M.; Lytinen, S.L.; Raison, K. Using game reviews to recommend games. In Proceedings of the Tenth Artificial Intelligence and Interactive Digital Entertainment Conference, Raleigh, NC, USA, 3–7 October 2014.
- Zagal, J.P.; Tomuro, N. The aesthetics of gameplay: A lexical approach. In Proceedings of the 14th International Academic MindTrek Conference: Envisioning Future Media Environments, Tampere, Finland, 6–8 October 2010; pp. 9–16.
- 17. Quadrana, M.; Cremonesi, P.; Jannach, D. Sequence-aware recommender systems. *ACM Comput. Surv.* (*CSUR*) 2018, 51, 66. [CrossRef]
- 18. Wan, M.; McAuley, J. Item recommendation on monotonic behavior chains. In Proceedings of the 12th ACM Conference on Recommender Systems, Vancouver, BC, Canada, 2–7 October 2018; pp. 86–94.
- 19. Hu, Y.; Koren, Y.; Volinsky, C. Collaborative filtering for implicit feedback datasets. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; pp. 263–272.
- 20. Rendle, S. Factorization machines with libfm. ACM Trans. Intell. Syst. Technol. (TIST) 2012, 3, 57. [CrossRef]
- 21. Davidson, J.; Liebald, B.; Liu, J.; Nandy, P.; Van Vleet, T.; Gargi, U.; Gupta, S.; He, Y.; Lambert, M.; Livingston, B.; et al. The YouTube video recommendation system. In Proceedings of the Fourth ACM Conference on Recommender Systems, Barcelona, Spain, 26–30 September 2010; pp. 293–296.
- 22. Koren, Y.; Bell, R. Advances in collaborative filtering. In *Recommender Systems Handbook*; Springer: Berlin, Germany, 2015; pp. 77–118.
- 23. Guo, Y.; Wang, M.; Li, X. An interactive personalized recommendation system using the hybrid algorithm model. *Symmetry* **2017**, *9*, 216. [CrossRef]
- 24. Salakhutdinov, R.; Mnih, A.; Hinton, G. Restricted Boltzmann machines for collaborative filtering. In Proceedings of the 24th International Conference on Machine Learning, Corvallis, OR, USA, 20–24 June 2007; pp. 791–798.

- 25. He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; Chua, T.S. Neural collaborative filtering. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 173–182.
- Sedhain, S.; Menon, A.K.; Sanner, S.; Xie, L. Autorec: Autoencoders meet collaborative filtering. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 111–112.
- Wu, Y.; DuBois, C.; Zheng, A.X.; Ester, M. Collaborative denoising auto-encoders for top-n recommender systems. In Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, San Francisco, CA, USA, 22–25 February 2016; pp. 153–162.
- Rendle, S.; Freudenthaler, C.; Schmidt-Thieme, L. Factorizing personalized markov chains for next-basket recommendation. In Proceedings of the 19th International Conference on World Wide Web, Raleigh, NC, USA, 26–30 April 2010; pp. 811–820.
- 29. He, R.; McAuley, J. Fusing similarity models with markov chains for sparse sequential recommendation. In Proceedings of the 2016 IEEE 16th International Conference on Data Mining (ICDM), Barcelona, Spain, 12–15 December 2016; pp. 191–200.
- Zhang, C.; Wang, K.; Yu, H.; Sun, J.; Lim, E.P. Latent factor transition for dynamic collaborative filtering. In Proceedings of the 2014 SIAM International Conference on Data Mining, Philadelphia, PA, USA, 24–26 April 2014; pp. 452–460.
- Wu, C.Y.; Ahmed, A.; Beutel, A.; Smola, A.J.; Jing, H. Recurrent recommender networks. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, Cambridge, UK, 6–10 February 2017; pp. 495–503.
- Mikolov, T.; Karafiát, M.; Burget, L.; Černocký, J.; Khudanpur, S. Recurrent neural network based language model. In Proceedings of the Eleventh Annual Conference of the International Speech Communication Association, Makuhari, Japan, 26–30 September 2010.
- Liu, Q.; Zeng, Y.; Mokhosi, R.; Zhang, H. STAMP: Short-term attention/memory priority model for session-based recommendation. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 1831–1839.
- Li, J.; Ren, P.; Chen, Z.; Ren, Z.; Lian, T.; Ma, J. Neural attentive session-based recommendation. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 1419–1428.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
- 36. Zhang, S.; Tay, Y.; Yao, L.; Sun, A. Next Item Recommendation with Self-Attention. *arXiv* 2018, arXiv:1808.06414.
- Xu, C.; Zhao, P.; Liu, Y.; Xu, J.; Sheng, V.S.S.; Cui, Z.; Zhou, X.; Xiong, H. Recurrent Convolutional Neural Network for Sequential Recommendation. In Proceedings of the 2019 World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 3398–3404.
- 38. Ying, H.; Zhuang, F.; Zhang, F.; Liu, Y.; Xu, G.; Xie, X.; Xiong, H.; Wu, J. Sequential recommender system based on hierarchical attention networks. In Proceedings of the 27th International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018.
- Koren, Y. Collaborative filtering with temporal dynamics. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 28 June–1 July 2009; pp. 447–456.
- 40. Bharadhwaj, H.; Joshi, S. Explanations for temporal recommendations. *KI-Künstliche Intell.* **2018**, *32*, 267–272. [CrossRef]
- 41. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* 2014, arXiv:1406.1078.
- 42. Marung, U.; Theera-Umpon, N.; Auephanwiriyakul, S. Top-N recommender systems using genetic algorithm-based visual-clustering methods. *Symmetry* **2016**, *8*, 54. [CrossRef]

- Pan, R.; Zhou, Y.; Cao, B.; Liu, N.N.; Lukose, R.; Scholz, M.; Yang, Q. One-class collaborative filtering. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; pp. 502–511.
- 44. Tseng, P. Convergence of a block coordinate descent method for nondifferentiable minimization. *J. Optim. Theory Appl.* **2001**, 109, 475–494. [CrossRef]



 $\odot$  2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).