Enriching Sequential Recommendations with Contextual Auxiliary Information

Adel Alkhalil
Department of Information and Computer Science, College of Computer Science and Engineering,
University of Ha'il, Ha'il, 81481, Saudi Arabia

***Abstract*—Recommender Systems (RS) play a key role in offering suggestions and predicting items for users on e-commerce and social media platforms. Sequential recommendation systems (SRS) leverage the user's previous interaction history to forecast the next user-item interaction. Although deep learning methods like CNNs and RNNs have enhanced recommendation quality, current models still face challenges in accurately predicting future items based on a user's past behavior. Transformer-based SRS have shown a significant performance boost in generating accurate recommendations by using only item identifiers which are not sufficient to generate meaningful and relevant results. These models can be improved by incorporating descriptive features of the items, such as textual descriptions. This paper proposes a transformer-based SRS, ConSRec, Contextual Sequential Recommendations, that incorporates auxiliary information of the items, such as textual features, along with item identifiers to model user behavior sequences for producing more accurate recommendations. ConSRec builds upon the BERT4Rec model by integrating auxiliary information through sentence representations derived from the textual features of items. Extensive experiments conducted on several benchmark datasets demonstrate substantial improvements compared to other advanced models.**

***Keywords*—Recommender system; sequential recommendation; auxiliary information; sentence transformer; sentence embedding**

I. Introduction

A Recommender system (RS) is designed to predict user preferences based on user intent, which can vary over time [2], [1], [3], [4]. To better capture users' dynamic preferences, several sequential recommendation (SR) techniques have been recently introduced. These methods utilize a user's past interaction history to predict the next item they are likely to engage with [7], [6], [5], [8], [9].

Earlier SR models relied on Markov Chain models to capture user preferences and predict the next item in the sequence [10], [6]. With the rise of deep learning, many SR models have transitioned to using neural network architectures like RNNs [12], [11] and CNNs [13]. Later, attention-based Transformer models [14], [32] were introduced to address SR problems, such as SASRec [8] employs a uni-directional attention mechanism, which limits its ability to capture comprehensive user preferences. BERT4Rec [9] addresses this by adopting a bi-directional transformer architecture to learn contextual relationships from both directions. However, both models rely primarily on item identifiers and fail to incorporate auxiliary contextual data, such as textual descriptions or reviews, which are critical for improving recommendation accuracy, especially under sparse data conditions.

KeBERT4Rec [15] incorporated keywords along with item identifiers in BERT4Rec model by concatenating the keyword representation with items by using one-hot encoding to generate the keyword vector, which did not capture the contextual meaning of keywords. Although, these SR models show significant performance gain, however, they do not exploit contextual features to generate meaningful representations.

Incorporating such contextual auxiliary information into SR models not only improves recommendations, particularly under sparse situations but also has significant practical implications in real-world applications. For instance, in e-commerce, enriching recommendations with textual features like product descriptions can enhance personalized shopping experiences, leading to increased user satisfaction and higher sales conversion rates. Similarly, in media streaming services, leveraging textual metadata such as genre descriptions or user reviews can better align recommendations with user preferences, enhancing user engagement. To achieve this, we propose a model called ConSRec, Contextual Sequential Recommendations, a modification of the BERT4Rec model, which includes contextual descriptions. The proposed model, ConSRec, leverages a transformer-based architecture to incorporate contextual auxiliary information, such as textual descriptions and user reviews, into sequential recommendation tasks. This design makes it highly effective in addressing key challenges like sparse user-item interactions and the inability of existing models to utilize rich contextual data. By integrating multi-head self-attention mechanisms, ConSRec ensures the effective fusion of sequential and contextual data, providing a robust solution to improve recommendation accuracy. The major objectives of the paper can be outlined as below:

- Introduce a model that incorporates the user sequences with contextual item descriptions using Masked Language Model (Close Objective Task) and bidirectional transformers.
- Generate meaningful representations using contextual description of the items using Sentence-BERT.
- Evaluation and performance comparison of the model against existing leading models.

The article is structured as follows: Section II provides a review of the relevant literature, while Section III offers a detailed explanation of the proposed model. Section IV covers the experimental evaluation of the work, and Section V concludes the paper.

II. Literature Review

Sequential recommendation system is a type of RS that exploits the user interaction sequences to infer the successive

item [17]. The aim of SR is to recommend future product by considering historical behavior of users. This historical behavior is also known as next item prediction. Earlier, the SRS were introduced using Markov Chains (MC) models for capturing sequential patterns from the user historical preferences [18], [10], [19]. The next item preferred by the users are predicted depending upon the last item, thus interpreting only the adjoining sequential behavior.

Recurrent Neural Network (RNN) based models exploiting Gated Recurrent Unit (GRU) [20], [12] and Long Short-Term Memory (LSTM) [39] have showed substantial performance gain for SR [25], [21], [22], [11], [12], [24], [23]. RNNs enforce rigid sequential patterns for encoding user preferences for making predictions. Besides RNN, a number of Convolutional Neural Network (CNN) [26], [27] based RS have also been introduced that also target problems related to the sequential recommendation. For example, Tang and Wang *et al.* [13] exploit CNN for capturing local sequential features using more recent behaviors.

Recently, Transformer models based on the attention mechanism [14] have achieved outstanding results in various deep learning tasks, including text classification [28], image captioning [29], and machine translation [30]. Initially developed for natural language processing, Transformers have also transformed the field of sequential recommendation (SR) [17] by leveraging the encoder component to process sequential data. This converts the sequence of items, representing the user's interaction history, into a sequence of vector representations [14]. The input sequence of items is first embedded and then concatenated with positional embeddings capturing the item's position in the sequence.

TABLE I. COMPARISON OF SR MODELS

Model	Pros	Cons
GRU4Rec	GRU model of RNN with ranking-based loss function.	Aimed for session-based recommendation systems using RNN.
SASRec	First unidirectional self-attention sequential model based on Transformers for next-item recommendation.	Uses only left-to-right attention, limiting its ability to learn hidden representations bidirectionally.
BERT4Rec	Bidirectional model employing Transformer architecture with multi-head self-attention to analyze user behavior sequences.	Lacks incorporation of additional information to produce meaningful predictions.
KeBERT4Rec	Extension of BERT4Rec leveraging keywords alongside item identifiers for next-item prediction.	Keyword representations are not derived using contextual embedding techniques.
FDSA	Segregated attention blocks exploit items and their features to predict the next item.	Heterogeneous item characteristics make it challenging to determine user preferences accurately.
S3Rec	Self-supervised model utilizing attribute data to learn correlations among items.	Does not use descriptive information for generating meaningful representations.

Most of recent sequential models shown in Table I follows the transformer architecture comprising of encoder block [31], [8], [9], [15] and using the item identifiers for next item recommendation. A feature level deeper self attentive model [16] introduced by T. Zhang *et al* exploits segregated attention blocks for items and their associated features to predict next item. In [33] proposed S3Rec, a self supervised SR model that utilized the attribute data of item to learn the correlation among them. KeBERT4Rec [15] leverages the keyword by integrating them with item identifier for the prediction of next item in sequence.

Existing sequential recommendation models, such as SASRec and BERT4Rec, rely primarily on item identifiers and focus on implicit feedback for next-item prediction. SASRec's uni-directional attention mechanism limits its ability to fully capture sequential dependencies, while BERT4Rec's bi-directional architecture, although more robust, still neglects auxiliary information like textual descriptions and user reviews. These omissions reduce the models' effectiveness in scenarios with sparse data or ambiguous user-item interactions, which necessitate richer contextual representation. To address this we proposed ConSRec, a model that combines auxiliary information and item identifiers to create embeddings using the Sentence BERT [34] embedding technique. This enhances item recommendation and prediction accuracy by capturing contextualized representations.

III. METHODOLOGY

The suggested framework "ConSRec - Contextual Sequential Recommendation System" is depicted in Fig. 1. The proposed paradigm is developed based upon Transformer architecture that adapted the deep bidirectional BERT model for SR prediction task (Fig. 2).

Fig. 1. Proposed methodology outline diagram.

A. System Overview

Before passing the sequence of items to the model proposed, the auxiliary features of these items are taken as input to the Sentence-BERT. This auxiliary information is the textual description of the items in the form of sentences that are processed to extract the contextual dense feature representation. These dense embedding are extracted prior to training phase to reduce the model training time.

Subsequently, during the training process, the auxiliary information's embeddings of items within a sequence are extracted and then passed to the embedding layer. These embeddings are subsequently combined with the item's embedding and positional embedding to capture the sequential behavior of the items. Only the encoder part of Transformer is used to compute the hidden representation using self attention mechanism for each item.

This resultant concatenated item's representations of sequence are then processed through stack of Transformer layer from [14] where hidden features for each item are calculated simultaneously at each layer. These layers share information bidirectionally across each position in hierarchical manner. After processing through all layers, a final learned hidden representation is projected at output layer that contemplated the future item recommendation for a user.

Several experiments were carried out on three benchmark datasets—MovieLens-1M, MovieLens-20M, and Amazon Beauty—to validate the effectiveness of the proposed model. The model's architecture includes an embedding layer, a transformer layer, and an output layer.

B. Mathematical Formulation of Proposed Model

Let set of users be shown mathematically as $\mathcal{U}$ = $\{u_1, u_2, u_3, ..., u_{|\mathcal{U}|}\}$ $\mathcal{M}$ = $\{m_1, m_2, m_3, ..., m_{|\mathcal{M}|}\}$ be the set of items. For each item, there is some item description (auxiliary information) that is in textual form denoted as $\mathcal{TD}$ = $\{des_1, des_2, des_3, ..., des_{|\mathcal{M}|}\}$. The items interacted in the sequence $\mathcal{S}$ in historical order for a user u is denoted as $\mathcal{S}$ = $\{m_1, m_2, m_3, ..., m_n\}$ where m_n is a particular item from $\mathcal{M}$, the user has acted upon previously. Given the sequence history $\mathcal{S}$, the objective of the SRS is to anticipate the future item m_n+1, the model will predict as:

$$\mathcal{P}(m_n\text{+1} = m|\mathcal{S})$$

C. Embedding Layer

The recommendation model in [9] makes use of the positional embedding along with the item identifier embedding to maintain the sequence of the items, thus memorizing the sequential order of the input. However, the pair alone does not describe the contextual representation of the input. It also does not recommend contextually especially under sparse conditions. To overcome this limitation, ConSRec incorporates additional auxiliary information based on contextualized description of items. The model utilizes the Sentence-BERT [34] for capturing contextual representation of the item descriptions. The architecture of Sentence-BERT for extracting sentence embedding is depicted in Fig. 3.

Sentence-BERT first utilizes BERT to generate word/ token embedding. Input in the form of sentences or text of various length is injected to the selected SBERT model, that generates contextualized word embedding for all input tokens in the sentence. Secondly, these word embedding are passed through a pooling layer to generate a fixed-sized vector representation. Among various pooling options available, the model utilizes the mean pooling in which mean of all contextualized token embedding is calculated to produce a fixed dimensional output embedding vector. Given the item descriptions of various length of all items, $\{\mathcal{TD}\}$ as input, the model produces 384 dimensional dense vector representation $\{Emb_{\mathcal{TD}}\}$ as in Eq. 1. These 384 dimensional embeddings are then used along with the item identifier and position embedding to produce information rich vector representations as shown in Eq. 2.

$$SBERT(\{\mathcal{TD}\}) = \{Emb_{\mathcal{TD}}\} \tag{1}$$

In the proposed model, *d* dimensional embedding layer is constructed by summing up the item identifier embedding, the position embedding and the additional auxiliary information (item description) extracted from $\{Emb_{\mathcal{TD}}\}$. Thus, for a given item m_i, the input embedding matrix $\mathcal{EM}$ is formulated by adding the corresponding item embedding E_m, position embedding E_{pos} and textual description embedding E_{des} as:

$$\mathcal{EM}_m = E_m + E_{pos} + E_{des} \tag{2}$$

D. Transformer Layer

The summed embedding $\mathcal{EM}$ becomes the input to the transformer layer that iteratively calculates the hidden representations of each item at each layer.

The structure of transformer layer or simply the encoder layer is build using the "multi-head attention". The layer piles up multiple encoder blocks [14] each consisting of "Multi-Head Self Attention" sub-layer and a "Position-wise Feed Forward Network". Given that $\mathcal{E}^{\curlywedge} = [\mathcal{EM}_{\curlywedge}^{\curlywedge}, \mathcal{EM}_{\curlywedge}^{\curlywedge}, ...\mathcal{EM}_{\approx}^{\curlywedge}]$ depict the dense vector embedding of the user sequence to the transformer, multi head self attention layer, MHSA is defined as:

$$MHSA(\mathcal{E}^l) = [h_1; h_2; ....h_h]W^0 \tag{3}$$

$$h_i = Attn(\mathcal{E}^l W_i^Q, \mathcal{E}^l W_i^K, \mathcal{E}^l W_i^V) \tag{4}$$

where W_i^Q, W_i^K and $W_i^V \in \mathbb{R}^{dxd/h}$ are the three learnable projection weight matrices and $W_i^0 \in \mathbb{R}^{dxd}$. $\mathcal{E}^l W_i^Q$, $\mathcal{E}^l W_i^K$, $\mathcal{E}^l W_i^V$ are the three linear transformation of input vector representation $\mathcal{E}'$ for Query, Key and Value (Q,K,V) vectors. Here, the attention function is scaled dot product [14] computed as:

$$Attn(Q, K, V) = \sigma\left(\frac{QK^T}{\sqrt{d/h}}\right)V \tag{5}$$

where the Query, Key and Value matrices are denoted by Q, K, V respectively and σ is the softmax function. Let MHSA at the l^{th} layer be S_i. Since, the MHSA block is based on linear projections, thus, the non-linearity to the MHSA is empowered by applying position-wise feed-forward network layer, PFN on all MHSA(S_i) separately.

$$PFN = [FNL(S_1^l)^T, FNL(S_2^l)^T .....FNL(S_n^l)^T] \tag{6}$$

$$FN(S_i) = GELU\big(S_i W^{(1)} + b^{(1)}\big)W^{(2)} + b^{(2)} \tag{7}$$

A smoother GELU activation function is used inline with BERT [5] and OpenAI GPT [40]. $W^{(1)}, b^{(1)}, W^{(2)}$, and $b^{(2)}$ are hyper-parameters communicated at all layers. Complexity of the model is reduced using residual connection at each sub layer. Dropout is applied followed by layer Normalization, LNorm. Thus, the sub-layer output at each level is $LNorm(x + Dropout(sublayer(x)))$. Input at each layer is denoted by x in the LNorm and represented as:

$$\mathcal{E}^l = Trm\big(\mathcal{E}^{l-1}\big), \quad \forall i \in [1, 2, ...L] \tag{8}$$

$$whiteA = Dropout\big(PFN(S_i^{l-1})\big) \tag{9}$$

$$Trm(\mathcal{E}^{l-1}) = LNorm\big(S_i^{l-1} + A\big) \tag{10}$$

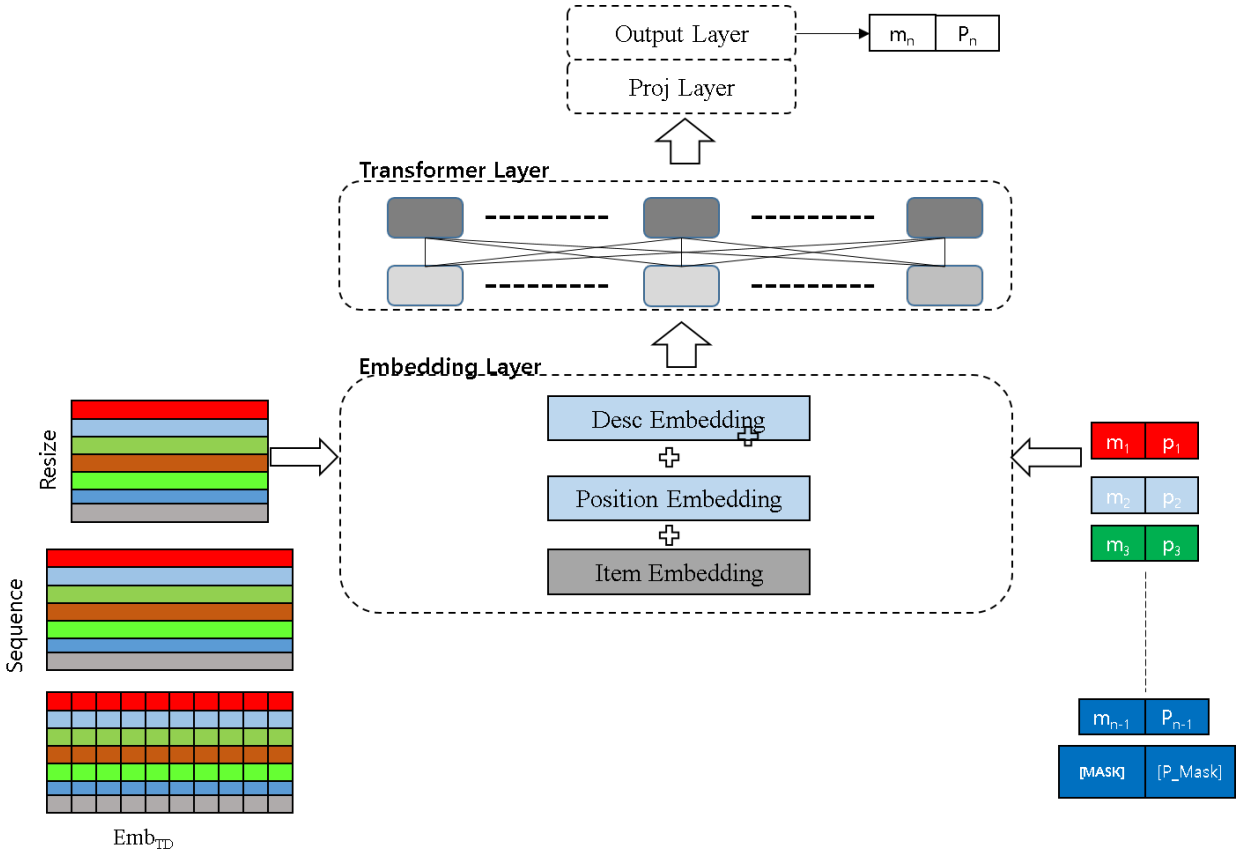

Fig. 2. Model architecture of contextual sequential RS.

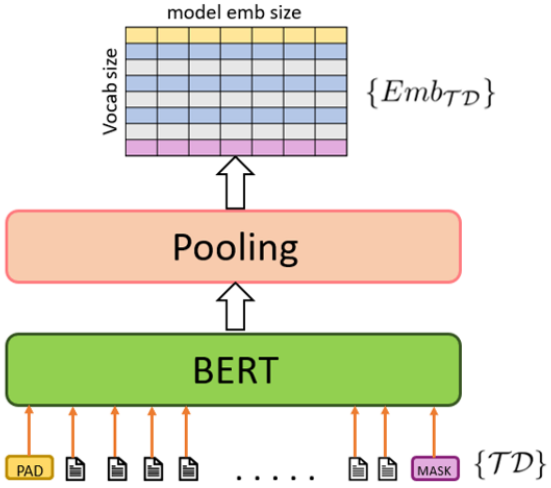

Fig. 3. Design architecture of sentence-BERT.

$$whiteB = Dropout\big(MHSA(\mathcal{E}^{l-1})\big) \tag{11}$$

$$S_i^{l-1} = LNorm\big(\mathcal{E}^{l-1} + B\big) \tag{12}$$

E. Output Layer

After passing through L layers and shared representations bidirectionally over each position in hierarchical manner, a final learned hidden representation $\mathcal{E}^{\mathcal{L}}$ is projected at output layer for all input item sequences. Considering the last item m_n in the sequence is masked, m_n is anticipated using embedding sequence $\mathcal{E}^{\mathcal{L}}$ that is depicted in Fig. 2. The last layer applies linear transformation twice followed by softmax function to predict the masked item.

$$whiteG = GELU\big(\mathcal{E}_t^L W^P + b^P\big) \tag{13}$$

$$P(m) = softamx\big(G(\mathcal{E}\mathcal{M}^T) + b^O\big) \tag{14}$$

where b^P and b^O are the bias at projection layer and W^P is the projection matrix. $\mathcal{E}\mathcal{M}^T$ is the item embedding matrix comprising of item identifier, positional and auxiliary information embedding. Here, shared item embedding is applied to minimize model size and relieve over-fitting.

IV. Experimental Configurations and Results Analysis

The datasets used to evaluate the proposed model and their preparation followed by experiment setup, evaluation metrics and performance comparison are presented in this section.

Most of the State-of-the-art SRS are trained on benchmark datasets that includes MovieLens, Amazon – Beauty. To effectively compare the improvements in the proposed model, these models are chosen. Moreover, auxiliary information of these datasets are also available and can easily be incorporated in these datasets. The auxiliary information, movie plot summary, of MovieLens dataset has been obtained through IMDbPy and for the Beauty Dataset, the description is chosen as a auxiliary information which is extracted from the meta file associated with the beauty dataset. Any item lacking auxiliary information has been ignored.

A. Dataset Pre-processing

Performance of proposed model is demonstrated through experiments carried out on three benchmark datasets including movielens-1m , movieLens-20m (ml-1m[1] and ml-20m[2]) and Amazon-Beauty[3] as described below:

1) MovieLens: A well-known dataset most commonly used for evaluating the performance of SRS. MovieLens ratings dataset contains the user id, item id (IDs of the movies from "movies" table), ratings and timestamp for movie ratings from each user. The auxiliary information (movie plot summary) for MovieLens is extracted through IMDbPY[4] using the **ImdbId** unique identifier, thus, making it information rich dataset.

2) Amazon - Beauty: It is a set of dataset comprising of reviews of a number of products extracted from "Amazon.com". The data is split into multiple datasets based upon product categories on Amazon. In our experiments, "Beauty" category is chosen that has a "rating" and a "meta" file. To incorporate the auxiliary information in the "rating" dataset, "description" of each product is extracted from the "meta" dataset.

The following Table II summarizes the dataset statistics.

TABLE II. DATASETS STATISTICS

Dataset	#users	#items	#interactions	Sparsity
ML-1m	6,040	3,706	1.0m	95.16%
ML-20m	138,493	26,744	20.0m	95.53%
Beauty	22,363	12,101	0.2m	99.93%

B. Evaluation Metrics

To measure the overall SR behavior, widely used leave-one-out strategy [8], [9], [15] is employed. The last item in each user's sequence is used for testing for every user, the second-to-last item is used for validation, and the remaining interaction items are utilized for training. For fair assessment, commonly used sampling practice [8], [9], [15] i.e. the ground truth object is coupled item with 100 negative items in test set based on how popular they are.

For evaluating all methods, "Normalized Discounted Cumulative Gain" (NDCG), "Hit Ratio" (HR) and "mean reciprocal rank" (MRR) are calculated. Higher values of these metrics depicts how better the recommendation performance is. Hit Ratio (HR) is used for measuring the ranking accuracy by comparing the test item set (T) with the ranked list. Mathematically it is expressed as:

$$HR@K = \frac{Numberof Hits@K}{|T|} \tag{15}$$

HR@K calculates the number of hits in a K-sized list. A hit occurs if the item tested is available in ranked list. Whereas the relative position of that item is assessed using NDCG in the ranked list. It assigns higher scores if the item is present at top position in the list. Mathematically it is evaluated by following formula:

$$NDCG@K = N_K \sum_{j=1}^{K} \frac{2^{z_j} - 1}{log_2(j+1)} \tag{16}$$

where N_K is the normalizer and z_j being the item's graded relevance at position j. We compute both the metrics of every test user items and then take their mean.

C. Baselines

For performance comparisons, we consider the following methods.

1) BPR-MF: [35] This model is the first one that uses the Bayesian personalized ranking loss for the optimization of matrix factorization (MF).

2) NCF: [36] This model utilizes MLP for capturing the item sequence interacted by user instead of using inner product in MF.

3) FPMC: [6] Combines MF with first-order MC to capture the long-term preferences of the user.

4) GRU4Rec: [12] It models the user click sequences using RNN-GRU for session based recommendation.

5) SASRec: [8] It is a unidirectional (left-to-right) self attentive model for next item prediction.

6) BERT4Rec: [9] This top of the line model uses bidirectional self attentive blocks and Cloze [37] masking for the recommendation task.

7) KeBERT4Rec: [15] This model extends BERT4Rec [9] by integrating keywords as additional input layer.

D. Implementation Details

The proposed model is trained on machine having 16 GB RAM and NVIDIA GTX 3080Ti (11GB). The training of proposed model is done using Adam Optimizer [38] with initial learning rate lr) of 0.001 and weight decay of 0.01. The hidden dimension is set to 128, dropout to 0.1 and 200 value used for maximum sequence length for MovieLens datasets and 50 value for Amazon Beauty. The masking probability of 0.15 is set for ML-20m and ML-1m. A 256 of batch size is used to train the proposed model.

The code provided by the corresponding authors of the respective baselines models were executed on the same machine. The optimized settings for hyperparameter values are used for all baseline models. The *hidden dimensionality* is tested from {64,128,256}, *dropout* from {0.1-0.9}, l_2 *regularizer* from {0-0.0001}.

[1] https://grouplens.org/datasets/movielens-1m/
[2] 1https://grouplens.org/datasets/movielens-20m/
[3] 3http:/jmcauley.ucsd.edu/data/amazon/
[4] https://imdbpy.github.io/

E. Comprehensive Performance Analysis

Tables III, IV and V presents the optimized outcomes of each baseline models on benchmark datasets. The highest scores in each table are shown in bold, while the 2nd place scores are underlined. The last row in each table displays how the proposed model performs in comparison to the best baseline model. The advantage of FPMC over BPR-MC is that it sequentially models users' previous records. From this observation, the importance of taking sequential pattern in consideration for recommendation systems can be ascertained.

Comparing the sequential baseline models, SASRec model outperforms GRU4Rec and FPMC on all benchmark datasets. This observation demonstrate that use of transformer based self attention models are more accurate than using traditional mechanisms. However, SASRec performance fall behind as compare to BERT4Rec which depicts that bidirectional model like BERT4Rec is more powerful as compared to unidirectional model like SASRec. BERT4Rec is a SR model that relies only on the item identifiers for the purpose of generating representation/ embedding. This model ignores the auxiliary information that is already provided with the datasets. However, KeBERT4Rec, a variant of BERT4Rec, has modified the representation by adding keywords describing the items e.g. Genre of movie.

TABLE III. Comprehensive Performance Analysis of Proposed Model with Referenced Models for Next Item Recommendations on ML-1m Dataset

ML-1m				
Metric	**HR@5**	**HR@10**	**NDCG@5**	**NDCG@10**
BPR-MF	0.2866	0.4301	0.1903	0.2365
NCF	0.1932	0.3477	0.1146	0.1640
FPMC	0.4297	0.5946	0.2885	0.3439
GRU4Rec	0.4673	0.6207	0.3196	0.3627
SASRec	0.5434	0.6629	0.3980	0.4368
BERT4-Rec	0.5876	0.6970	0.4454	0.4818
KeBERT4-Rec	0.5873	0.7651	0.5134	0.5488
ConSRec	**0.6690**	**0.7761**	**0.5287**	**0.5633**
Improvement	13.91%	1.44%	2.98%	2.64%

TABLE IV. Comprehensive Performance Analysis of Proposed Model with Referenced Models for Next Item Recommendations on ML-20m Dataset

ML-20M				
Metric	**HR@5**	**HR@10**	**NDCG@5**	**NDCG@10**
BPR-MF	0.2128	0.3538	0.1332	0.1786
NCF	0.1358	0.2922	0.0771	0.1271
FPMC	0.3601	0.5201	0.2239	0.2895
GRU4Rec	0.4657	0.5844	0.3091	0.3637
SASRec	0.5727	0.7136	0.4208	0.4665
BERT4-Rec	0.6325	0.7473	0.4967	0.5340
KeBERT4-Rec	0.8770	0.9450	0.7250	0.7470
ConSRec	**0.9863**	**0.9981**	**0.7687**	**0.8237**
Improvement	12.46%	5.62%	6.03%	10.27%

TABLE V. Comprehensive Performance Analysis of Proposed Model with Referenced Models for Next Item Recommendations on Beauty Dataset

Beauty				
Metric	**HR@5**	**HR@10**	**NDCG@5**	**NDCG@10**
BPR-MF	0.1209	0.1992	0.0814	0.1064
NCF	0.1305	0.2142	0.855	0.1124
FPMC	0.1387	0.2401	0.0902	0.1211
GRU4Rec	0.1315	0.2343	0.0812	0.1074
SASRec	0.1934	0.2653	0.1436	0.1633
BERT4-Rec	0.2207	0.3025	0.1599	0.1862
KeBERT4-Rec	0.3751	0.4753	0.2841	0.3164
ConSRec	**0.3884**	**0.5321**	**0.3261**	**0.3394**
Improvement	3.55%	11.95%	14.78%	7.27%

TABLE VI. Analysis on the Incorporation of Auxiliary Information

Model	**Metrics**	**BERT4Rec**	**ConSRec***	**ConSRec**
Beauty	**HR@10**	0.3025	0.3321	0.4631
	NDCG@10	0.1862	0.1922	0.3120
	MRR	0.1701	0.1653	0.2581
ML-1m	**HR@10**	0.6970	0.7023	0.7761
	NDCG@10	0.4818	0.4953	0.5633
	MRR	0.4254	0.4308	0.4484

TABLE VII. Impact of Using Contextual Embedding Technique

Dataset	**Metric**	**One-Hot Encoding**	**Word2Vec**	**Doc2Vec**	**SBERT**
Ml-1m	HR@1	0.3502	0.3601	0.3643	0.3672
	HR@5	0.6563	0.6589	0.6607	0.6690
	HR@10	0.7651	0.7690	0.7740	0.7761
	NDCG@5	0.5134	0.5198	0.5203	0.5287
	NDCG@10	0.5488	0.5590	0.5619	0.5633
	MRR	0.4322	0.4381	0.4443	0.4484
Beauty	HR@1	0.1897	0.1906	0.2012	0.2038
	HR@5	0.3432	0.3671	0.3874	0.3884
	HR@10	0.4983	0.5012	0.5296	0.5321
	NDCG@5	0.3079	0.3160	0.3256	0.3261
	NDCG@10	0.3187	0.3251	0.3361	0.3394
	MRR	0.2263	0.2476	0.2509	0.2517

The addition of this keywords embedding in the model makes KeBERT4Rec perform better than BERT4Rec. Thus, suggesting that incorporating some kind of side information along with item can improve the recommender's performance. It is evident from Table III that outcomes of all the sequential models like GRU4Rec, BERT4Rec, SASRec etc outperformed the non-sequential models like BPR-MC and NCF on dataset ml-1m. Our model outperformed in all baseline metrics showing the accuracy of model by incorporating the additional auxiliary information.

Result depicted in Table IV also indicates the importance of using Sentence-BERT to incorporates the contextual meaning of addition auxiliary information alongwith the item identifiers. Our model outperforms all baseline models. ConSRec gains an improvement of 5.62% on HR@10 and 10.27% on NDCG@10.

In accordance with the results, on the beauty dataset, Table V shows that our proposed model, ConSRec clearly

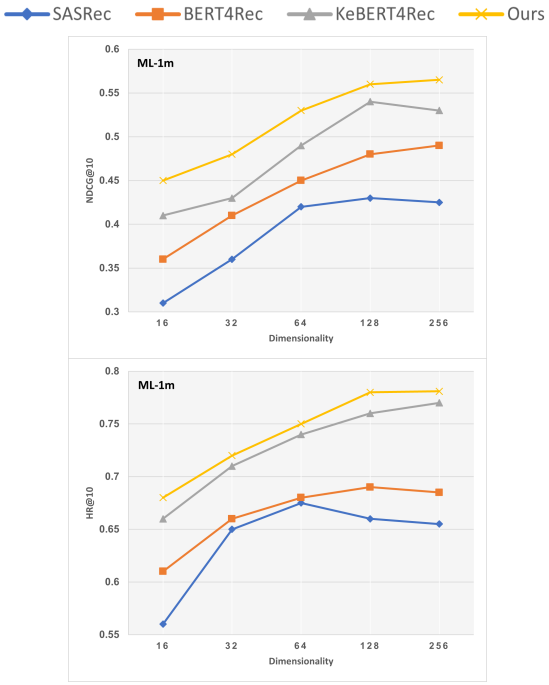

Fig. 4. Hidden dimensionality, d impact on NDCG@10 and HR@10 for ml-1m.

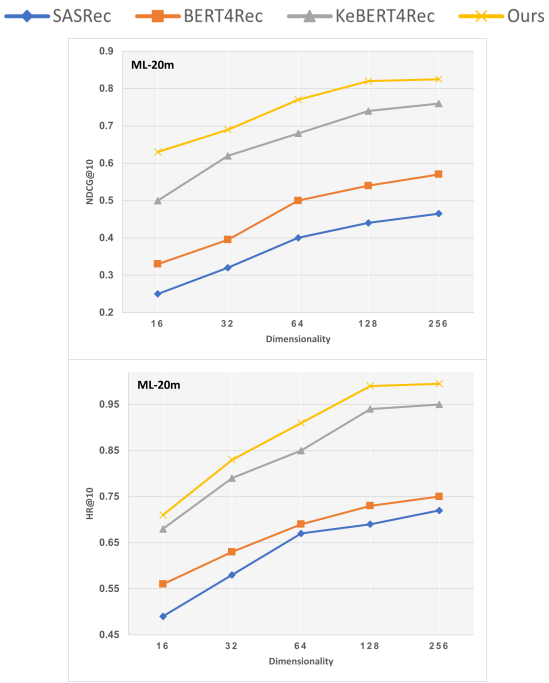

Fig. 5. Hidden dimensionality, d impact on NDCG@10 and HR@10 for ml-20m.

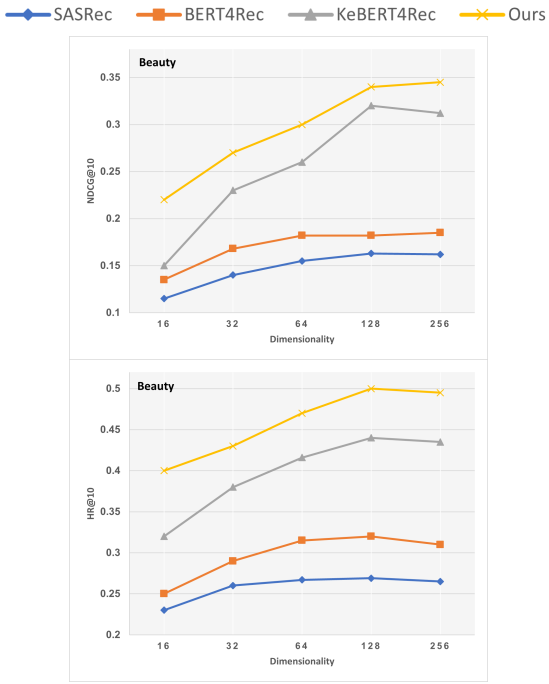

Fig. 6. Hidden dimensionality, d impact on NDCG@10 and HR@10 for beauty.

outperforms all baseline methods. The proposed model gains an average improvement of 11.95% on "HR@10" and 7.27% on "NDCG@10" as compared to the best baselines.

F. Evaluating Effect of "Hidden Dimensionality"

The hidden dimensionality *d* has a great impact on the performance of recommendation system that is studied in this section. Fig. 4, 5 and 6 exhibits the values of HR@10 and NDCG@10 on different baseline sequential model by varying hidden dimensionality *d* ranges between 16,32,64,128,256. The remaining of the hyperparameters are constant and kept to their optimal values.

The performance of ml-20m on varying dimensionality has very less impace on NDCG@10 and HR@10 as depicted in Fig. 5. However, bigger value of hidden dimensionality doesn't always yield accurate results.

It is obvious from Fig. 6 that with the increase of dimensionality, the graph of each model converges. However, improved model performance is not always achieved with bigger value of hidden dimensionality, particularly on sparse datasets, such as Beauty.

G. Impact of Integrating Auxiliary Information

As stated earlier, BERT4Rec only incorporated the embeddings of item identifiers along with its positional embedding in the input layer. However, in the proposed model, these embeddings are further enhanced and improved by incorporating additional contextual embedding layer of auxiliary information which is in the form of sentences.

The embeddings of this auxiliary information is extracted through Sentence-Transformer model that generates contextualized word embedding for all input tokens in the sentence.

To visualize this impact of using contextual information along with item identifier, the proposed model is initially trained by excluding the contextual information (ConSRec*). The results are then compared with the ConSRec i.e. by integrating side information. It is evident from the results that auxiliary information can enhance the productivity of SR system. Only the results on ml-1m and beauty dataset with batch size 128 are reported above in Table VI due to space limitations which clearly depicts that excluding the side information from proposed model degrades the performance.

H. Ablation Study

Finally, to visualize the impact of incorporating auxiliary information, some ablation experiments were conducted. Sentence-Transformer is used to train the proposed model which is a pre-trained model for generating embedding of item's side information. To analyze the impact of using contextual embedding instead of traditional techniques, the proposed model is tested using one hot encoding techniques to generate textual embedding.

As depicted in Table VII, By the use of Sentence Transformer to generate embeddings, the results of proposed model on ml-1m and beauty datasets outperforms all other non-contextual methods like word2vec, doc2vec, etc. This also emphasize the use of meaningful and context embedding generating technique for model training to produce relevant results.

V. Conclusion

Self Attention and Transformer based recommendation system have proven to be more precise and accurate as compared to traditional RS. In this paper, a transformer based sequential RS have been proposed that enhances recommendation accuracy by incorporating contextual auxiliary information of items in a sequence. The paper also uses contextual auxiliary information of items, such as descriptions or reviews, to enhance the recommendations. A contextual based pre-trained model sentence-transformer is used to generate meaningful embedding of auxiliary information. The experiments on various datasets show significant improvements over the state-of-the-art models.

However, ConSRec reliance on textual features as well as generalizability across highly diverse datasets beyond the domain of e-commerce and media streaming services poised limits to its capability. In future integration of additional multimodal data sources to further improve its performance and robustness.

References

[1] Zhiwei Liu, Mengting Wan, Stephen Guo, Kannan Achan, and Philip S Yu. 2020. Basconv: aggregating heterogeneous interactions for basket recommendation with graph convolutional neural network. In *Proceedings of the 2020 SIAM International Conference on Data Mining.* SIAM, 64–72.

[2] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *SIGIR.* 165–174.

[3] Liu, Zhiwei, et al. 2021. "Contrastive self-supervised sequential recommendation with robust augmentation." In *arXiv preprint arXiv:2108.06479.*

[4] Latifi, Sara, Dietmar Jannach, and Andrés Ferraro. 2022. "Sequential recommendation: A study on transformers, nearest neighbors and sampled metrics." In *Information Sciences* 609 (2022) : 660-678.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR abs/1810.04805*

[6] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, 2010. "Factorizing personalized markov chains for next-basket recommendation," *in Proceedings of the 19th International Conference on World Wide Web, ser. WWW '10.* ACM, 2010, p. 811–820.

[7] J. Tang and K. Wang, 2018. "Personalized top-n sequential recommendation via convolutional sequence embedding," *in Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining,* ser. WSDM '18. ACM, 2018, p. 565–573.

[8] W.-C. Kang and J. McAuley, 2018. "Self-attentive sequential recommendation," in *2018 IEEE International Conference on Data Mining (ICDM).* IEEE, 2018, pp. 197–206.

[9] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, ser. CIKM '19. ACM, 2019, p. 1441–1450

[10] Guy Shani, David Heckerman, and Ronen I Brafman. 2005. An MDP-based recommender system. *Journal of Machine Learning Research* 6, Sep (2005), 1265-1295.

[11] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent Neural Networks with Top-k Gains for Session-based Recommendations. In *Proceedings of CIKM.* ACM, New York, NY, USA, 843–852

[12] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," *in 4th International Conference on Learning Representations*

[13] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *Proceedings of WSDM.* 565–573.

[14] Vaswani, et. al. 2017 *Attention Is All You Need.* 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA.

[15] Elisabeth Fischer, Daniel Zoller, Alexander Dallmann, and Andreas Hotho. 2020. Integrating Keywords into BERT4Rec for Sequential Recommendation. In *KI 2020: Advances in Artificial Intelligence*.

[16] T. Zhang, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, D. Wang, G. Liu, and X. Zhou. 2019. Feature-level Deeper Self-Attention Network for Sequential Recommendation. In *IJCAI 2019. 4320–4326*.

[17] Aleksandr Petrov and Craig Macdonald. 2022. A Systematic Review and Replicability Study of BERT4Rec for Sequential Recommendation. In *Proc. RecSys*

[18] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th International Conference on Data Mining (ICDM).* IEEE, 191–200.

[19] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web.* 811–820.

[20] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of EMNLP. Association for Computational Linguistics,* 1724–1734.

[21] Qiang Liu, Shu Wu, Diyi Wang, Zhaokang Li, and Liang Wang. 2016. Contextaware sequential recommendation. In *2016 IEEE 16th International Conference on Data Mining (ICDM).* IEEE, 1053–1058.

[22] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In *Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 30*

[23] Tim Donkers, Benedikt Loepp, and J"urgen Ziegler. 2017 Sequential User-based Recurrent Neural Network ecommendations. In *Proceedings of RecSys,* ACM, New York, NY, USA, 152–160.

[24] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks. In *Proceedings of RecSys. ACM, New York, NY, USA, 130–137*

[25] Qiang Cui, Shu Wu, Qiang Liu, Wen Zhong, and Liang Wang. 2018. MV-RNN: A multi-view recurrent neural network for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering 32, 2 (2018),* 317–331.

[26] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. 2019. A simple convolutional generative network for next item recommendation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 582–590.

[27] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential Recommendation with User Memory Networks. In *Proceedings of WSDM. ACM, New York, NY, USA, 108–116.*

[28] Juyong Jiang, Jie Zhang, and Kai Zhang. 2020. Cascaded Semantic and Positional Self-Attention Network for Document Classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 669–677.

[29] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, 2015. "Show, attend and tell: Neural image caption generation with visual attention," in *ICML.*

[30] Jianling Wang, Kaize Ding, Liangjie Hong, Huan Liu, and James Caverlee. 2020. Next-item Recommendation with Sequential Hypergraphs. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1101–1110.

[31] Kyeongpil Kang, Junwoo Park, Wooyoung Kim, Hojung Choe, and Jaegul Choo. 2019. Recommender system using sequential and global preference via attention mechanism and topic modeling. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1543–1552.

[32] Potter, Michael, Hamlin Liu, Yash Lala, Christian Loanzon, and Yizhou Sun. 2022. "GRU4RecBE: A Hybrid Session-Based Movie Recommendation System *(Student Abstract).*"

[33] Zhou, Kun, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. "S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization." In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 1893-1902.

[34] Reimers, Nils, and Iryna Gurevych. 2019. "Sentence-bert: Sentence embeddings using siamese bert-networks." *arXiv preprint arXiv:1908.10084 (2019).*

[35] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, 2009. "BPR: bayesian personalized ranking from implicit feedback," in *UAI, 2009*.

[36] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of WWW. 173–182.*

[37] Wilson L. Taylor. 1953. Cloze procedure: a new tool for measuring readability. Journalism & Mass Communication Quarterly 30 (1953), 415–433.

[38] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." In *Proceedings of ICLR*.

[39] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. Neural Computation 9, 8 (Nov. 1997), 1735–1780.

[40] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. *Improving language understanding by generative pre-training.* In OpenAI Technical report.