Advanced Recommendations System using Hierarchical Recurrent Neural Network

1st Md. Anonto Shuvo
Dept. of CSE
BRAC University
Dhaka, Bangladesh
md.anonto.shuvo@g.bracu.ac.bd
ID: 23141036

2nd Md. Sifat Kamal
Dept. of CSE
BRAC University
Dhaka, Bangladesh
md.sifat.kamal@g.bracu.ac.bd
ID: 20101231

3rd Kazi Shahed Mamun
Dept. of CSE
BRAC University
Dhaka, Bangladesh
kazi.shahed.mamun@g.bracu.ac.bd
ID: 20301471

4th Sania Azhmee Bhuiyan
Dept. of CSE
BRAC University
Dhaka, Bangladesh
sania.azhmee.bhuiyan@g.bracu.ac.bd

5th Abid Hossain
Dept. of CSE
BRAC University
Dhaka, Bangladesh
abid.hossain@g.bracu.ac.bd

6th Annajiat Alim Rasel
Dept. of CSE
BRAC University
Dhaka, Bangladesh
annajiat@gmail.com

***Abstract*—The field of recommendation systems has seen significant advances due to the prevalence of complex data structures and the need for personalized user experiences. Traditional recommendation models often have difficulty capturing complex relationships in sequential and hierarchical data, which limits their ability to make accurate recommendations. We suggest a recommender system by proposing an "Advanced Recommendations System using Hierarchical Recurrent Neural Network" (HRNN). The proposed HRNN pushes the boundaries of traditional recommendation systems by addressing the challenges of scarcity, issues that start to cool, and changing user preferences. To evaluate the effectiveness of the advanced recommendation system using a hierarchical regression neural network, extensive tests were conducted on many real-world datasets. Benchmarks are performed based on conventional recommendation algorithms, highlighting the superiority of HRNN in terms of accuracy, recall, and user satisfaction metrics. This approach to recommender systems, harnessing the capabilities of hierarchical and recurrent neural networks to provide advanced, context-aware recommendations. This abstraction of the complexity of data is poised to redefine user experience in various domains, heralding a new era of intelligent and adaptive recommendation systems.**

***Index Terms*—Recommendation systems, Sequence Modeling, HRNN, RNN, Sequential Recommendation**

I. Introduction

According to the axiom of differentiability and continuity, a function must be continuous at every point within its domain in order for it to be differentiable at a point x 0. Additionally, all of the parameters used in the back propagation. must have continuous values. The situation might not always be the same, thus we might prefer a network made up of neurons that make challenging stochastic decisions about temporal occurrences at various time scales, such as a 0/1 value signifying the end of a video's narrative or the beginning of a text's word or phrase boundaries. These binary outputs also bring sparse representations that might be used as a regularization method, enabling the creation of gating units to determine which portion of the model actually has to be computed for a particular instance. [1] Thus, back propagation fails. In many online systems that use recommendations, interactions between users and systems are organized into sessions. A session is a series of interactions that take place within a particular environment time frame. Recurrent Neural Networks (RNNs) have been recently used for the purpose of sessionbased recommendations [2] outperforming item-based methods by 15% to 30% in terms of ranking metrics. In these cases it is reasonable to assume that the user behavior in past sessions might provide valuable information for providing recommendations in the next session. A novel algorithm based on RNNs that can deal with both cases: (i) session-aware recommenders, when user identifiers are present and propagate information from the previous user session to the next, thus improving the recommendation accuracy, and (ii) session-based recommenders, when there are no past sessions (i.e., no user identifiers). The algorithm is based on a Hierarchical RNN (HRNN) where the hidden state of a lower-level RNN at the end of one user session is passed as an input to a higher-level RNN [3]

II. RELATED WORKS

In this paper, Qu et al. [4] proposed a new RNN-based user behavior prediction model (RNN-BPM) to analyze different behavior sequences within online content consumption systems (e.g., data sparsity/cold-start/temporal context in recommendation systems). The RNN-BPM is based on the RNN language model and combines RNNs (Recurrent Neural Networks) and BPNNs (Backpropagation Neural Networks) to capture time-related and recurrent patterns in behavior. In the proposed RNNCM, RNNs are used to predict behavior by taking into account temporal context and repeating patterns. At the same time, three different types of behavior sequences i.e long-term behavior, short-term behavior and popular behavior are analyzed simultaneously. The authors compute the initial probabilities of probable items that a user would consume next time under various types of user behaviors using RNNs. They then used BPNN to build their BPNNCM model. BPNN calculates the final probability for each item that a user would likely consume next time based on the initial probabilities of those three behavior sequences. The RNN and BPNN models are nested to build the RNN-PBNCM, which predicts a user's next consumption. The experiments were performed on the Last.fm music app dataset and the results showed that the RNN – BPNNCM model was more effective than the baseline models and the single RNN model at predicting user behavior in the online content consumption system.

In this paper, Rosa et al. [5] presented a knowledge-based recommendation system (KBRS) which is designed to monitor the emotional health of users through an online social network (OSN). The system uses sentiment analysis and ontology to detect users with underlying psychological problems such as depression and stress. If detected, the system sends appropriate messages to the user based on their emotional state.The system uses convolutional neural network (CNN) and bidirectional long-term memory (BLSTM) - recurrent neural network (RNN) to classify sentences with depression or stress content. Test results show high accuracy in detecting users suffering from depression and stress. The recommended eSM2 Sentiment Index takes into account user profiles, including age, gender, geographic location, and subject, to provide more personalized sentiment analysis. The metric was developed based on subjective tests performed with participants assessing the sentimental value of sentences. KBRS architecture involves user profiles, messaging, depression/stress detection through machine learning, sentiment analysis using eSM2, ontology, and recommendation engine. The system was tested and compared with traditional RS without sentimental or ontological analysis. The proposed KBRS achieved significantly higher user satisfaction rates than traditional RS, demonstrating the effectiveness of using sentiment analysis and personalized ontology.

In this paper Choe et al. [6] presented a new recommendation system that considers item usage sequence as well as time intervals in the user history. The system is based on a Hierarchical Recurrent Neural Network (HRTN) model that has two layers: one for short-term events and one for long-term events. The first layer deals with short-term events, while the second layer carries long-term information over a larger time range, taking into account the time intervals in between events. They conducted experiments with real-world data sets, including Movielens data sets and Steam data sets, to see how well the proposed Hi-RNN model performed compared to baseline recommendation methods (i.e., RNN based models) and Hit@k (a metric that measures how accurately a top-k recommendation is made). Experiments showed that the Hi-RNN model performed significantly better when considering longer sequences, suggesting that temporal properties play an important role in recommendation systems (e.g., sequence length, time interval, etc.). The authors suggested that future work might include additional information, such as user trust relationships, review content, etc., to further improve the Hi-RNN model.

In this paper, Liu et al. [7] presented a new proposed model called Rating Graph Neural Network (RGNN) which aims to improve the accuracy of the recommendation system by using user reviews. The model builds a rating graph for each user/item, capturing the relationship between the words in the review. RGNN uses a type-aware graph attention mechanism and custom graph clustering operators to extract hierarchical semantic representations of the revision graph. Factoring machine (FM) class is used to predict user ratings based on learned semantic characteristics. The proposed RGNN is compared with various proposed methods based on the most modern assessments using real-world datasets. The test results demonstrate that RGNN consistently outperforms these methods in terms of mean squared error (MSE), showing its effectiveness in capturing important patterns in user reviews to have better suggestions. The authors also discussed the effects of different edge types in the rating graph, performed sensitivity analysis on hyperparameters, and presented a case study showing how RGNN selects information from the assessments to make accurate predictions.

In the paper Zhang et al. [8] presented a new recommendation framework named Sema, which is designed to improve personalized recommendations by combining semantic meaning and temporal dynamics using deep learning techniques. This framework uses hierarchical and symmetric recurrent neural networks (RNNs) to model semantic meaning and temporal dynamics for users and items. The authors tested on real-world datasets from Amazon and Yelp, and compared SEMA with some of the top recommended methods. The results demonstrate that SEMA achieves significant improvements in mean absolute error (MAE) and root mean square error (RMSE) compared with other methods. Overall, SEMA presents a comprehensive recommendation framework that effectively combines semantic analysis, time dynamics, and deep learning to improve personalized recommendations.

In the paper, Xu et al. [9] presented a new approach called Slander Detection Recommendation System (SDRS) to solve the problem of identifying slanderous users in referral systems. These are users who intentionally give misleading reviews and ratings to manipulate the system. SDRS uses a framework that combines sentiment analysis, user behavior analysis, and recommendation techniques to detect and manage these users. The authors identify the problem of detecting slanderous users and propose a comprehensive framework consisting of four main modules - integrate words, sentiment analysis to evaluate, detect and recommend slanderous users. Their goal was to identify slanderous users based on the difference between their ratings and reviews, and then improve the accuracy of the recommendation system by removing their influence. They conducted tests on many different datasets, including Amazon, Yelp, Taobao and Jingdong. It compared SDRS with different baselines, demonstrating its effectiveness in detecting slanderous users and improving the accuracy of recommendations. The results show that SDRS is superior to other sentiment and recommendation analysis methods in identifying these users and making accurate recommendations.

In this paper, Donkers et al. [10] presented a new approach to generate personalized recommendations using recurrent neural networks (RNNs) with a focus on the recommendation systems domain. Main idea of the authors was to integrate user-specific information into the RNN architecture to improve the quality of recommendations. The documentation provides three different variations of user-based RNNs - linear user integration, edited linear user integration, and attention user integration. These variations are designed to effectively incorporate user characteristics and preferences into the recommendation process. The results of offline tests performed on real-world datasets (MovieLens and LastFM) to compare suggested user-based RNN variants with baseline recommendation algorithms. various, including KNN based on matrix elements and coefficients. The rating metrics used are MRR@20 (Mean Reciprocal Rating) and Recall@20. Tests show that user-based RNN consistently outperforms baseline on two measures, demonstrating the effectiveness of integrating user information into sequence-based recommendations. The user-attention-integrated variant of the user-based RNN seems to be the most successful among the proposed variants. It can significantly improve the accuracy of recommendations compared to standard RNNs and other benchmarks. The paper ends by suggesting future research directions, including integrating the most advanced recommendation techniques and combining time intervals between events for more accurate predictions.

In the paper, Ko et al. [11] proposed a new approach for the recommender system using the online user's activity sequence. The authors introduced a cooperative sequence model based on recurrent neural networks (RNNs) to capture temporal patterns in user behavior. Unlike traditional models that focus on ratings, this approach looks at user action sequences and aims to predict future actions. Model characteristics include flexibility for different tasks, the ability to manage long-term dependencies, and a balance between personalization and collaboration aspects. This method is evaluated against music recommendation and mobile prediction tasks using real data sets, demonstrating its superiority over other methods. The authors suggested incorporating time information for future improvement.

In the paper, Song et al. [12] presented HCRNN (Hierarchical Contextual Recurrent Neural Network), a new model for sequential recommender systems. HCRNN aims to address challenges such as long-term dependencies and changing user behavior over time. The model introduces three hierarchical contexts - global, local and temporary benefits. It modifies the traditional LSTM structure to accommodate these contexts, incorporates triggering mechanisms to capture changes in interest, and utilizes dual-channel attention for efficient use of segmentation contexts. Different versions of HCRNN were introduced (HCRNN-1, HCRNN-2, HCRNN-3), each with specific modifications. The authors used the CiteULike, LastFM and MovieLens datasets for the tests, comparing HCRNN with several baseline models. The results indicate that HCRNN consistently outperforms the baseline models in terms of recall and average reciprocal ratings. The highest performing version is the HCRNN-3 with dual channel attention. The document also provides qualitative insights, such as context integration, portal analysis, and case studies, to better understand model behavior and performance.

In the paper, Villatel et al. [13] addressed the challenge of building effective recommender systems by evaluating Recurrent Neural Network (RNN) models for predicting user preferences over short and long time periods. The authors explored both short-term and long-term recommendation tasks, considering the nuances of user interactions within different time horizons. They proposed enhancements to RNN models and conducted experiments to assess their impact on performance. The authors compared the performance of the RNN model with the results of experiments performed on a real-world dataset (Yoochoose) and on an internal dataset (an internal dataset). The RNN models were compared against several baselines (POP, Item- KNN, CoEvent MF). The experimental results show the RNN model outperforms the baselines. The best configuration for the RNN model is the stacked GRU, which includes layer normalization, a tied item embedding, and a tied-item matrix.

III. HIERARCHICAL RECURRENT NEURAL NETWORK

Hierarchical Recurrent Neural Network (HRNN) is a class of stacked RNN models designed to model hierarchical structures in sequential data (text, video streams, speech, programs, etc.). Our HRNN model is based on RNN in the following way. (i) Adding a GRU layer to model information across user sessions and track the evolution of user interests over time. (ii) Using a powerful user-parallel mini-batch mechanism for efficient training.

A. Architecture

There are two GRU layers in HRNN in contrast to one layer in RNN [3]: (i) Session-level which is similar to the RNN's GRU layer. (ii) User-level GRU. The session-level GRU models the user activity within sessions and generates recommendations. The user-level GRU models the evolution of the user across sessions and provides personalization capabilities to the session-level GRU by initializing its hidden state. In this way, the information relative to the preferences expressed by the user in the previous sessions is transferred to the session-level GRU.

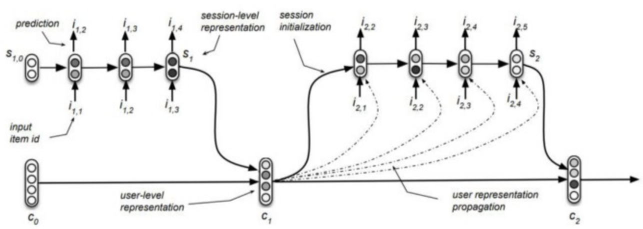

Fig. 1. Graphical representation of the proposed Hierarchical RNN model for personalized session-based recommendation

The following two different HRNN settings are considered depending on whether the user representation cm is considered:

• HRNN Init, in which cm is used only to initialize the representation of the next session.

• HRNN All, in which cm is used for initialization and propagated in input at each step of the next session.

In HRNN Init, the session-level GRU can exploit the historical preferences along with the session-level dynamics of the user interest. HRNN All instead enforces the usage of the user representation at session-level at the expense of a slightly greater model complexity. As it will be seen, this can lead to substantially different results depending on the recommendation scenario.

B. Training

There are various methods for training such networks: (a) Gradient estimation techniques, that train binary neurons as an integral component of the network, [1] (b) Policy gradientbased reinforcement learning methods [14] that bypass the necessity for binary neurons by instead creating a map of stateaction pairs to decide on what action to take on observing a particular input. These can be conducted by two classes of HRNN namely Hierarchical Multiscale Recurrent Neural Networks (HM-RNN) & Hierarchically Structured LSTMs (HS-LSTMs). Moreover, (c) With user-parallel mini-batches [15] we can train HRNNs efficiently over users having different number of sessions and sessions of different length. Moreover, this mechanism allows to sample negative items in a user-independent fashion, hence reducing the chances of 'contamination' of the negative samples with actual positive items. The sampling procedure is still popularity-based, since the likelihood for an item to appear in the mini-batch is proportional to its popularity

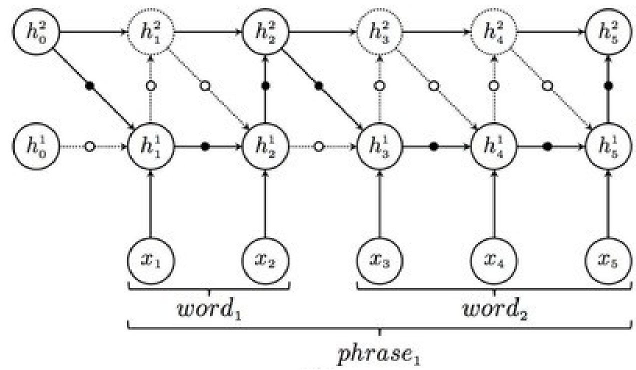

Fig. 2. Hierarchical Multiscale Recurrent Neural Networks (HM-RNN)

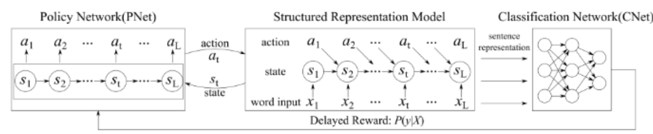

Fig. 3. Hierarchically Structured LSTMs (HS-LSTMs)

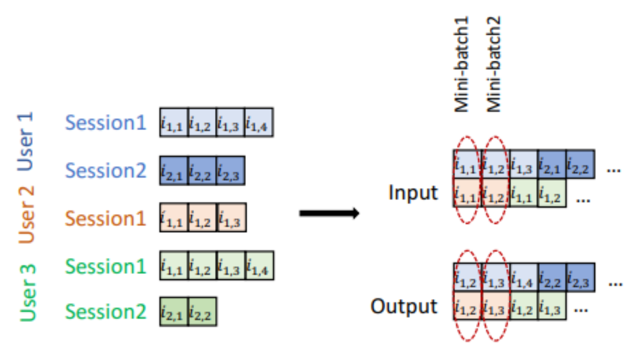

Fig. 4. User-parallel mini-batches for mini-batch size

IV. Experiments

A. Dataset

We are going to use 4 publicly available real-world datasets which are Movielens1, CiteULike, Steam and LastFM. We are going to compare two types of RNN models with the baseline models and also use the dataset for each model separately. The two types of RNN models are Hierarchical Recurrent Neural Network (H-RNN) and Hierarchical Context Enabled Recurrent Neural Network (HC-RNN). Both of these are variations of recurrent neural networks that are designed to capture hierarchical structures and dependencies in sequential data. In general there is not enough difference between these two models as as both are Hierarchical Recurrent Neural Network.

For HRNN, we going to use Movielens1 and Steam data-set. In the Movielens1 dataset, there are 6040 users and 3416 items. The average sequence length per user is 163.5.

[6] Moreover, Steam is the game platform dataset containing 334730 users and 13047 items. The average sequence length per user is 12.26. [6] For test dataset, most recent item of each user was used. For verification, the 2nd most recent item was used and the rest for training.[6] From both the datasets, users with less than 5 events was filtered out. On the other hand, for HC-RNN, Movielens1, CiteULike and LastFM data-sets are chosen. Sequence having length less than 10 are removed. Also item which are less than 50/50/25 each in the 3 different dataset. Cross-validation was performed by selecting 10% of the randomly chosen training set as the validation set. [12] The dataset has five part. These are sequence(train), sequence(test), clicks, items, average length. Sequence(train) has 38,724 CiteULike data, 73,420 LastFM data and 136,233 Movielens1 data. Sequence(test) has 9,140 CiteULike data, 17,829 LastFM data and 34,682 Movielens1 data. Click has 1,163,813 CiteULike data, 4,575,159 LastFM data and 5,041,882 Movielens1 data. Items has 1980 CiteULike data, 5778 LastFM data and 930 Movielens1 data. Average length of has CiteULike data is 24.31, LastFM data is 50.14 and Movielens1 data is 29.50. [12]

B. Baselines

Baseline Models are some common models that is used in recommendation system. We are going compare these baseline models with the proposed models. Some baseline models are-

- **POP:** It is one of the most popular. this model recommends items based on their popularity, without considering any user-specific information. It simply suggests the most popular items to all users, which can be a simple but often effective baseline.
- **SPOP (Session-Based Most Popular):** It is an extension of the POP model that focuses on recommending popular items within a user's current session or recent browsing history, making it more suited for session-based recommendations.
- **ITEM-KNN (Item-Based k-Nearest Neighbors):** It is a collaborative filtering model that suggests items to a user based on the similarity of items they have interacted with in the past. It calculates item-item similarity and recommends items that are most similar to those the user has engaged with.
- **BPR (Bayesian Personalized Ranking):** It is a matrix factorization model that optimizes the ranking of items in a personalized way. It uses pairwise ranking to learn user and item latent factors, aiming to rank items that the user has interacted with higher than those they haven't.
- **BPR-MF (BPR Matrix Factorization):** It is an extension of the BPR model that specifically uses matrix factorization techniques to learn user and item embeddings for personalized recommendations.
- **GRU4REC (Gated Recurrent Unit for Recommendations):** it is a session-based recommendation model that uses Gated Recurrent Units (GRUs) to capture sequential patterns in user interactions within a session and recommends items accordingly.
- **GRU4Rec+ (Enhanced GRU4Rec):** It is an improved version of the GRU4REC model, incorporating additional techniques such as negative sampling and adaptive learning rates to enhance recommendation performance.
- **LSTM4REC (Long Short-Term Memory for Recommendations):** Similar to GRU4REC, it uses Long Short-Term Memory (LSTM) networks to capture sequential behaviors and make personalized recommendations within sessions.
- **NARM (Neural Attentive Session-based Model):** It is a session-based recommendation model that employs neural attention mechanisms to focus on important items in a user's session history while making recommendations.
- **STAMP (Session-based Temporal Attention Model with Personalization):** It is a session-based recommendation model that combines temporal attention mechanisms with personalization, considering both sequential patterns and user preferences.
- **FMC (Factorization Machines for Context-Aware Recommendation):** It is a model that utilizes factorization machines to capture interactions between user preferences and contextual features, enabling context-aware recommendations.
- **FPMC (Factorized Personalized Markov Chains):** It is a hybrid model that combines matrix factorization and Markov chain techniques to capture both user-item interactions and sequential dependencies for improved recommendations.

We are going to compare HRNN with POP, BPR, FMC, FPMC, GRU4Rec and GRU4Rec+ baseline models. On the other hand, HC-RNN with POP, SPOP, Item-KNN, BPR-MF, GRU4Rec, LSTM4REC, NARM AND STAMP.

REFERENCES

[1] J. Chung, S. Ahn, and Y. Bengio, "Hierarchical multiscale recurrent neural networks," *arXiv preprint arXiv:1609.01704*, 2016.

[2] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," *arXiv preprint arXiv:1511.06939*, 2015.

[3] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, "Personalizing session-based recommendations with hierarchical recurrent neural networks," in *proceedings of the Eleventh ACM Conference on Recommender Systems*, 2017, pp. 130–137.

[4] Z. Qu, S. Zheng, X. Wang, X. Song, B. Li, and X. Song, "Converged recommendation system based on rnn and bp neural networks," in *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2018, pp. 308–313. DOI: 10.1109/BigComp.2018.00052.

[5] R. L. Rosa, G. M. Schwartz, W. V. Ruggiero, and D. Z. Rodríguez, "A knowledge-based recommendation system that includes sentiment analysis and deep learning," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2124–2135, 2019. DOI: 10.1109/TII.2018.2867174.

[6] B. Choe, T. Kang, and K. Jung, "Recommendation system with hierarchical recurrent neural network for long-term time series," *IEEE Access*, vol. 9, pp. 72 033–72 039, 2021. DOI: 10.1109/ACCESS.2021.3079922.

[7] Y. Liu, S. Yang, Y. Zhang, C. Miao, Z. Nie, and J. Zhang, "Learning hierarchical review graph representations for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 1, pp. 658–671, 2023. DOI: 10.1109/TKDE.2021.3075052.

[8] J.-D. Zhang and C.-Y. Chow, "Sema: Deeply learning semantic meanings and temporal dynamics for recommendations," *IEEE Access*, vol. 6, pp. 54 106–54 116, 2018. DOI: 10.1109/ACCESS.2018.2871970.

[9] Y. Xu, Y. Yang, J. Han, E. Wang, J. Ming, and H. Xiong, "Slanderous user detection with modified recurrent neural networks in recommender system," *Information Sciences*, vol. 505, pp. 265–281, 2019.

[10] T. Donkers, B. Loepp, and J. Ziegler, "Sequential user-based recurrent neural network recommendations," in *Proceedings of the eleventh ACM conference on recommender systems*, 2017, pp. 152–160.

[11] Y.-J. Ko, L. Maystre, and M. Grossglauser, "Collaborative recurrent neural networks for dynamic recommender systems," in *Asian Conference on Machine Learning*, PMLR, 2016, pp. 366–381.

[12] K. Song, M. Ji, S. Park, and I.-C. Moon, "Hierarchical context enabled recurrent neural network for recommendation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, 2019, pp. 4983–4991.

[13] K. Villatel, E. Smirnova, J. Mary, and P. Preux, "Recurrent neural networks for long and short-term sequential recommendation," *arXiv preprint arXiv:1807.09142*, 2018.

[14] T. Zhang, M. Huang, and L. Zhao, "Learning structured representation for text classification via reinforcement learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.

[15] B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk, "Parallel recurrent neural network architectures for feature-rich session-based recommendations," in *Proceedings of the 10th ACM conference on recommender systems*, 2016, pp. 241–248.