CEDTECH International Journal of Engineering & Applied Science
Volume 2, Number 3, September 2021
http://www.cedtechjournals.org
ISSN: 2756-4533

COMPARATIVE ANALYSIS OF AUTOENCODER AND LSTM IN MUSIC SESSION DATA

Agbator Oseremen Lawrence
Department of Computer Science
Edo State Polytechnic, Usen Edo State
Email: mail4agbator@gmail.com

ABSTRACT

Deep learning is now being used in the very complicated area of item recommendation, where each e-commerce site has its own quirks, thanks to the exponential increase in computing power and memory. The difficulty offered by data sparsity in this domain is immense, and it has had a significant impact on the performance of existing algorithms in this domain. Our research aimed to uncover a new method of eliciting latent factors implicit in the sequence of items consumed by a user over time by listening to a song or watching a movie. This is possible because a song playlist created by a user is indicative of the user's preference for the songs in the sequence in a given session. We examine the effectiveness of sequential LSTM and Autoencoder LSTM in learning the latent characteristics present in the session sequence of consumable online items, exhibiting the usefulness derived for enhanced recommender system.

Keywords: *Autoencoder-LSTM, Sequential-LSTM, Session, Item sequence, Latent-factors.*

INTRODUCTION

During the SPARK + AI summit 2018 in Europe, IBM's Nick Pentreath stated that recommendation systems are among the first systems to use machine learning in commercial applications in the real world. Recommendation and personalization systems have grown ingrained in our daily interactions, whether through search engines, social media platforms, or large and small e-commerce sites. According to Nick, this influence extends not just to the user experience during interactions, but also to the money generated by industry participants. Users are at the heart of recommendations; as they engage with goods, they generate a wealth of data insights that may be used to develop recommendation and personalization systems. Users can locate products of interest on a variety of platforms and in a variety of ways thanks to recommendation systems. This recommendation system, on the other hand, examines an individual user's previous activity in a user community, looking for patterns that disclose both implicit and explicit features of the user and a specific item. When these patterns are identified, they are separated to find combinations that meet the consumers' preferences (Quadrana et al, 2018).

Because the computational system basically predicts the user rating on items, or would-be users rating on items, and then uses this rating to predict scores used to rank this item in their perceived order of importance to the would-be user, recommendation systems can be safely described as ranking systems and also prediction systems. Items can be recommended in two ways: personalize and non-personalized. Personalize involves recommending a popular item to everyone in the same domain based on what other customers or users have said about it. As you might expect, this is usually a simple and straightforward activity that does not necessitate the use of innovative algorithms. The customized recommendation is the second category; this approach considers individual preferences and attempts to provide a personalized experience for the user (Phan & Le, 2019)

The personalised recommendation system is by far the most popular, and as a result, it is being used more widely in both academia and industry. Many algorithms have been used in this research, and the methods and the application domain are both evolving. Every year, researchers publish a large number of papers in this category, and many modifications to current algorithms and new algorithms are applied to this field of research, which also attracts a lot of industrial sponsorship. Collaborative filtering, content-based filtering, and the hybrid filtering technique are three of the most common approaches that have been researched in this area.

Items are recommended for users who are assumed to be similar based on their demographics or preferences derived from previous user item interactions in collaborative filtering. The object and the user are the entities of interest explored to build profiles used for similarity modeling in the neighborhood based collaborative filtering approach.

LITERATURE REVIEW

Data mining is the detection and extraction of insights and patterns from a huge collection of data sets in the field of information technology. One of the fastest-growing areas of data mining is recommendation systems, sometimes known as recommender systems. Today, this system has sparked a lot of interest in the industry and academia, owing to the rise of e-commerce. It aids commercial enterprises in developing personalized business strategies that help them provide quality service to their customers by presenting items and products that they are personally interested in, which helps to increase their profits(Lipi et al, 2016).

The discovery of recommendation systems is said to be the result of numerous research projects in the disciplines of cognitive science, information retrieval, common forecasting theories, and approximation theory. It is also thought that the field of recommendation systems began in the mid-1990s, and that today there are many applications that use recommendation systems(Lipi et al, 2016). Goldberg, Nichols, Oki, and Terry are thought to have created the first recommendation system in 1992. The Tapestry was an electronic rating system that allowed people to review goods and products based on their likes and dislikes. This was the first recognition system, and it was used to personalize the prediction of items and products for users, as well as to display a selection of products that had been identified as being of interest to them(Lipi et al, 2016). Various academics have conducted numerous studies in various areas of recommendation systems. Others are statistical modeling of latent features using a variety of methods, while others are utilizing diverse approaches to implement collaborative, content-based, and hybrid filtering. In the next session, an attempt will be made to list a few works.

The authors in (Shakila et al, 2017)said that in a graph-based approach, ecommerce sites and search engines do not incorporate sematic and sentiment properties, which they believe can improve recommendation by bridging a perceived information vacuum that exists in traditional collaborative and content-based filtering. As a result, they offer a system that integrates recommendation and semantics using the overlap semantic technique. Some of the qualities of their system include modeling of people and goods information held in the database, as well as ensuring that user input and ratings based on various criteria are considered. In their research, they discovered that incorporating semantic factors into a recommendation system can help to increase and improve the effectiveness of a recommendation system.

The author in (Yifan et al, 2018) used his movie recommendation research to examine three very straightforward and basic tactics that are typically overlooked: recency, similarity, and ratings. The techniques of recency and item similarity are non-personalized, whereas the approach of rating is customized. Three experimental algorithms and one baseline algorithm were devised and implemented in this study. Each of the three related item algorithms generates trailer recommendations based on a user's current viewing of a trailer. The researchers computed scores between pairs of movies using a content-based metric technique based on the similarity of latent feature vectors generated by a supervised machine learning procedure. We calculated predicted ratings using item-item collaborative filtering. The highest click through rate (CTR) was determined to be Film Release Date, followed by Tag Similarity.

In their agent-based intelligent social tagging system, the researchers in (An et al, 2017) used the social tagging system to improve how the required resources are recommended to users quickly and accurately. They used agent technology to build a system that mines user interests, performs personalized and common preference group recommendations, and proposes a self-adaptive recommendation strategy that achieves an equilibrium between efficiency and accuracy in both the personalized and common preference recommendation models. The distributed genetic algorithm can be used to improve this agent-based recommendation system (Inoue et al, 2016).

The researcher in (Frolov&Oseledets, 2016) investigated a collaborative filtering model based on a third order tensor factorization approach. They represented explicit feedback as an ordinal utility measure rather than a cardinal measure, giving their model sensitivity to both negative and positive user input and the capacity to offer information about all conceivable user preferences. In their investigation, they discovered that this methodology substantially simplified the learning of cold start user preferences and improved the quality of advice.

A hierarchical Bayesian strategy was presented in the study work of (Guigourès et al, 2018) to handle the problem of size suggestion in e-commerce fashion, which was viewed as a challenge in fashion recommendation. In their technique, they combined the size ordered by a client with the possibility of a return event (1. no return, 2. returned too small, 3. returned too big). The underlying parameters were obtained by mixing priors from a multinomial distribution based on the joint probability of each event in a hierarchical fashion.

In a similar study, (Fernández-Tobía&Tomeo, 2016) investigated multiple recommendation algorithms in the movie and music domains, both in single-domain and cross-domain scenarios, on a dataset with only positive input. They were able to show that positive feedback from cross-domain preference data is useful for providing more accurate suggestions when user feedback in the target domain is scarce or non-existent, and their evidence also showed that this approach has the potential to lead to more diverse recommendations when comparing the methods in terms of item ranking accuracy, diversity, and catalog coverage. A simultaneous joint modeling of the user's preferences at aligned sites can aid in capturing the user's overall global preferences. This can be extremely beneficial, especially in the case of a cold start (Cao &Yu, 2016).

Another tensor factorization-based technique is being investigated (Chou et al, 2016) took a content-based approach to his research. To overcome the cold start problem in next-song suggestion, this solution uses content features collected from music audio. Based on the last song a user listened to, the computer learns sequential behavior to forecast the next song the user will enjoy. During the iterations of the factorization process, our technique has the unique property of learning and updating the mapping between the audio feature space and the item latent space.To provide efficient cold start recommendations, the latent attributes for the user and products can be properly derived from the content features.

The changing nature of human taste poses a variety of issues in product and item selection. It has been observed that the utility of user-items can fluctuate in different contests in several settings. The context can include things like time, location, and the presence of others (Soaresý&Dologý, 2019). The researchers proposed collective embeddings for a recommender system that is aware of neural contests. To capture temporal patterns, the solution models the user, object, and time embeddings together. Using the outer product of the user-item-time correlation between the embedding space's dimensions. The non-linearity in this feature space is learned by a convolution neural network. The output from the CNN is used in the fusion layer to generate the user preference score (Soaresý&Dologý, 2019).

The researcher in (Twardowski, 2016) who employed RNN to analyze the sessional interactions of users in the setting of a search engine rather than an ecommerce site, discovered that users generate a lot of preferences during sessions in the search engine. A Session-Aware Recommender System technique is provided in the paper, which does not require any straightforward user information. Only user activity inside a single session, defined as a sequence of actions, is used in the recommendation process. Explicit context modeling using factorization methods and an unique methodology with Recurrent Neural Networks are used to include this information into the recommendation process (RNN) (Twardowski, 2016).

The authors in (Covington et al, 2016) described a deep neural network architecture for YouTube video recommendation that they divided into two problems: candidate creation and ranking. The collaborative filtering model was discovered to effectively integrate multiple signals and describe their interaction with depth layers, exceeding prior matrix factorization algorithms utilized at YouTube. Using the age of the training sample as an input feature removes an inherent bias towards the past and helps the model to capture the time-dependent behavior of famous films, according to their findings. In A/B testing, this enhanced offline holdout precision results and significantly increased watch time on recently uploaded videos. Other works (Kim et al, 2016) used Convolutional Matrix Factorization for Document Context-Aware Recommendation to overcome the problem of sparsity, which often makes systems that rely on user ratings inconvenient. They developed a method for improving recommendation accuracy when reviews are utilized instead of explicit ratings, a feature that has provided a significant issue in characterisation due to the bag of words model's intrinsic limits.

This assumption is further bolstered by (Barkan et al, 2019), who suggested a deep neural multiview model that serves as a bridge from item content to their CF representations. In cold start recommendation, (Qian et al, 2020) employed a variation of graph neural networks to outperform the baseline. They suggested a graph neural network that employs an attribute graph rather than a user-item graph, whose performance is dependent on the density of user-item interactions. Researchers have been exploring various ways to deal with the issue of latent factors that cannot be generated effectively when the degree of sparsity is over the roof or in scenarios of first time user or new item, (Roy&Chandra, 2016). Considered emotion modeling in their work Latent Factor Representations for Cold-Start Video Recommendation in the same effort to deal with the issue of latent factors that cannot be generated effectively when the degree of sparsity is over the roof or in scenarios of first time user or new item. The researchers attempted to bridge the gap between latent components and emotion modeling. The success of this project demonstrated that when latent components from user-item interactions are missing, emotion modeling steps in to save the day, and that emotional aspects play a significant role in linking users to items. This was shown, however, in their conclusions on how to interpret latent components in terms of the emotional component they contain.

METHODOLOGY

To apply a collaborative filtering technique in our song recommendation, we strive to employ a strategy that is different from the popular methodology in recommendation systems for the elicitation of hidden latent characteristics between users. We used a sequence of song IDs as input to our RNN LSTM layer in this study to predict an output sequence of songs where the first song ID in the output sequence is dependent on the first song ID in the input sequence. In our dataset, the output sequence is the input sequence minus the first song ID in the series.

Input sequence A = (1,2,3,4,5.6,7)

Output sequence B =(2,3,4,5,6,7)

The length of the output sequence LB is the length of the input sequence (LA)-1. As we refer to sequence here as an ordered list to be computed having a defined length where each song ID in any given list belongs to the entire datasets of songs, and all the ordered list also belongs to the set of sessions. For most work in this domain a list is of elements is defined for each user so you have that the predicted rank list Lr = f((u,s.i)**β**), the function is fed with user(u), session(s) and item(i) embeddings to predict the probability of items Lr in the output list. Our work takes only the ordered list representing the sequence of items which is song.

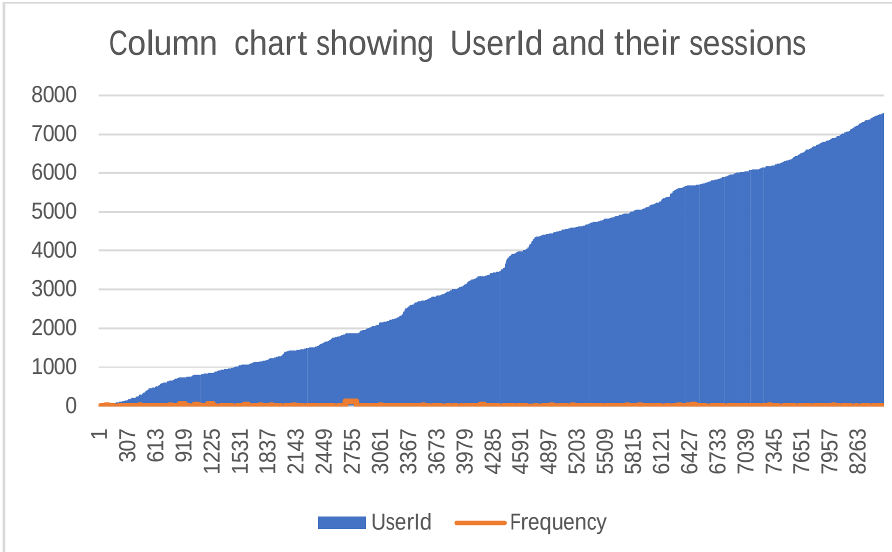
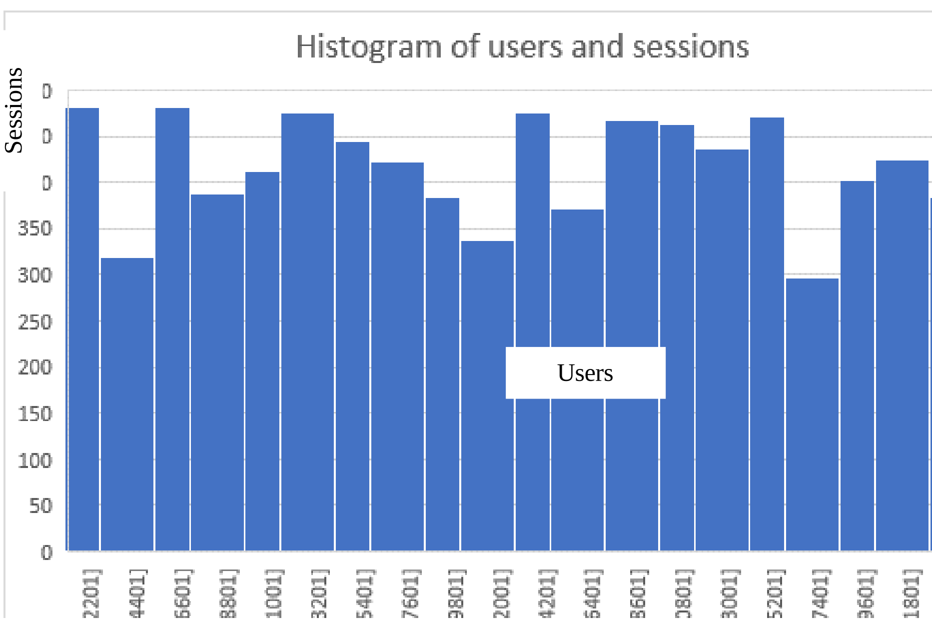


Fig. 1. User-Id and their Sessions

This chart showed that users with higher User Id had many sessions in the entire datasets. Visualizing on a histogram clearly showed the group of users that created the highest number of sessions as can be seen in the chart below.



The visualization of dataset set the tone for data preprocessing; we needed to know how many sessions each user had. There was concern that such a large gap would result in unbalanced data, which might not provide the necessary insight into the experiment. Because this imbalance will always exist in real life, the goal was to evaluate how much the results would be skewed by it and what kind of parameter tuning would be required to limit the effect while maintaining a good prediction.

The datasets consist of randomly distributed songs made by different users in different sessions; thus the initial step was to segregate the sessions and put them in a one-dimensional array so that each session could be fed into the network as a single sample. The next step was to format the output by removing the first song from the input and creating a proper situation in which one song predicts the following song.

The session data from the 30music dataset contained a variety of song counts, as one would expect given that a session playlist can contain any number of songs. As a result, there were now samples of varying lengths, so a temporary length must first be determined then truncate those that are longer than the length and pad the sessions that are shorter. Then, using the numpyhstack function, all of the samples were combined and molded into a three-dimensional array, which served as the input dimension for the LSTM layer.

Network architecture

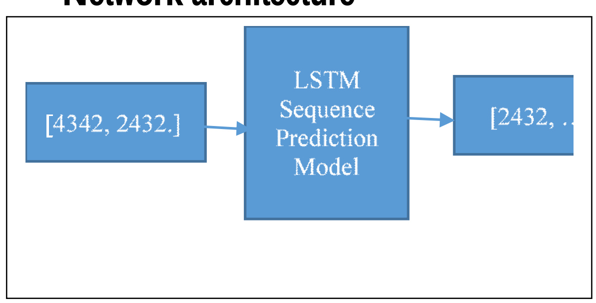


Figure 3 LSTM time series forcasting architecture

The above architecture in fig 3 shows the basic concept of the RNN architecture, from vanilla to LSTM all have the simple basic concept. In this work experiment was done with the sequential LSTM and LSTM autoencoder in our sequence-to-sequence problem which has witnessed poor performance with the vanilla RNN because of information morphing and vanishing or exploding gradient descent.

RESULTS AND DISCUSSION

The table below shows results from our experiment with sequential LSTM and LSTM autoencoder

Table 1 Results of slight variations of Parameters in sequential LSTM architecture

Validation loss	Optimizer	Epoch	Error function	Activation Function	No. of samples
0.0012	adam	100	mse	tanh	8000
0.0034	adam	100	mse	tanh	5000
0.0038	adagrad	100	mse	tanh	5000
0.0085	rmsprop	100	mse	tanh	5000
0.0003021	adam	5000	mse	tanh	5000
0.0238	adam	5000	mse	softmax	5000
0.00096302	adam	5000	mse	Relu	5000
0.0052	adam	5000	mse	sigmoid	5000

The table above shows the different performance of the parameters with the data and sequential LSTM, parameters tuning is a tricky operation, there is no way of knowing without experimentation what will be a better combination? The adam optimizer with tanh activation function and mean squared error function performs better in this experiment.

Table 2 Result of slight variations of parameters in LSTM Autoencoder Architecture

Validation loss	4.0366e-5	1.4849e-6	7.9333e-7
Optimizer	adam	adam	adam
Epoch	500	500	500
Error function	mse	mse	mse
Activation function	relu	relu	tahn
No of samples	80	1	80

This architecture also shows that the tanh activation function outperforms the relu activation function even when the epoch is reduced. In this experiment, the LSTM autoencoder obviously outperforms the sequential LSTM, and the performance is unaffected by the number of samples. This design learns the underlying relationship in a sequence quickly and may improve session-based recommendation performance in a real-world setting.

RESULT ADAPTATION

In this portion of the experiment, the nearest neighbour was used to reconstruct the output array by extracting items-Id closest to components of the output array using the output acquired from both LSTM architectures. This is necessary because, in the feature engineering, categorical features were not encoded into values that would reduce the overhead on processing resources; instead, it was only rescaled into continuous values that would be suitable for the LSTM cell to regress over time using the songs ID's time dependency. The performance of this model was equivalent to common baselines, according to the data summary from model evaluation.

Table 3 Mean Reciprocal Rank and Recall

Validation Loss	optimizer	epoch	Recall	MRR
0.0003021	adam	5000	0.4142	0.154
0.0005379	adagrad	5000	0.3541	0.152
	rmsprop	5000	0.3142	0.144

Table 3 confirms the prediction that if elicited by a deep learning network that performs back propagation through time, the hidden link and underlying similarities between the ordered elements in a session sequence are sufficient to create future recommendations. It is also seen that projected items were not always in their original place, which contributed to the low MRR. However, in this example, MRR is very important because all the items are part of the same playlist, and the user presumably likes all of the songs equally. The evaluation result was produced using the LSTM autoregression model, which outperformed the sequential model. In the sequential LSTM experiment, a dropout rate of 0.2 percent was maintained. It is e expected that finer parameter adjustment and a longer epoch will further improve this finding.

CONCLUSION

This research aims to uncover hidden factors in objects that a user interacts with in a sequential fashion, where the user interaction history is represented by merely a series of items in each session. Because of the exponential development in processing power and memory, deep learning's potential may now be effectively utilized in sequence-to-sequence prediction, a task that has proven problematic for traditional neural networks due to the limitations. We'd like to investigate how this can be applied to different domains and how parameters can be fine-tuned with less effort in the future.

REFERENCES

An, S. Zhao, Z.. Zhou, H(2017): Research on an Agent-based Intelligent Social Tagging Recommendation System. *2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics.*

Barkan O., Koenigstein N., Yogev E. and Katz O.(2019): CB2CF: A Neural Multiview Content-to-Collaborative Filtering Model for Completely Cold Item Recommendations. *In Proceedings of Recsys'19.ACM, Copenhagen, Denmark.* https://doi.org/10.1145/3298689.3347038

Cao, X. Yu, Y.(2016): Joint User Modeling across Aligned Heterogeneous Sites. *RecSys '16, September 15-19, 2016, Boston, MA, USA c 2016 ACM. ISBN 978-1- 4503-4035-9/16/09.* DOI: http://dx.doi.org/10.1145/2959100.2959155.

Chou, S. Yang, Y. Roger J. J,. Lin, Y. (2016): Addressing Cold Start for Next-song Recommendation. *RecSys '16, September 15–19, 2016, Boston, MA, USA.* DOI: http://dx.doi.org/10.1145/2959100.2959156.

Covington, P. Adams, J. Sargin, E.(2016): Deep Neural Networks for YouTube Recommendations: *RecSys '16 September 15-19, 2016, Boston , MA, USA.* DOI: http://dx.doi.org/10.1145/2959100.2959190

Fernández-Tobías, I. Tomeo, P.(2016): Accuracy and Diversity in Cross-Domain Recommendations for Cold-start Users with Positive-only Feedback. *RecSys '16, September 15-19, 2016, Boston , MA, USA.* DOI: http://dx.doi.org/10.1145/2959100.2959175

Frolov, E. Oseledets, I.(2016): Fifty Shades of Ratings: How to Benefit from a Negative Feedback in Top-N Recommendations Tasks. *RecSys '16, September 15–19, 2016, Boston, MA, USA.* DOI: http://dx.doi.org/10.1145/2959100.2959170.

Guigourès, R. King Y. Ho, Koriagin, E.(2018): A Hierarchical Bayesian Model for Size Recommendation in Fashion. *RecSys '18, October 2–7, 2018, Vancouver, BC, Canada.*

Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2016). Session-based Recommendations with Recurrent Neural Networks. *CoRR, abs/1511.06939*.

Inoue, M. Takenouchi H. and Tokumaru M.(2016): Music recommendation system improvement using distributed genetic algorithm: *2016 Joint 8th International Conference on Soft Computing and Intelligent Systems and 2016 17th International Symposium on Advanced Intelligent Systems.*

Kim D., Park C., Oh J., Lee S. and Yu, H.(2016): Convolutional Matrix Factorization for Document Context-Aware Recommendation. *RecSys'16, September 15-19, 2016, Boston, MA, USA.* DOI: http://dx.doi.org/10.1145/2959100.2959165.

Lipi S., Hetal G., Prem B. (2016): Survey on Recommendation System. *International Journal of Computer Applications* (0975 – 8887) Volume 137 – No.7, March 2016.

Ms. Shakila S. Dr. Sheetal R, Asst Prof. Prachi J. (2017): Recommendation system in E-commerce websites: A Graph Based Approached. *2017 IEEE 7th International Advance Computing Conference.*

Pentreath, N. (2018): SPARK + AI summit in Europe https://databricks.com/speaker/nick- pentreath

Phan D. H, Le D. H. (2019). E-Commerce Recommendation System Using Mahout. *IEEE 4th International Conference on*

Computer and Communication Systems.

Qian T., Liang Y., Li Q. (2020): Solving Cold Start Problem in Recommendation with Attribute Graph Neural Networks. :arXiv:1912.12398 [cs.IR] 2020

Quadrana, M. Cremonesi, P. and Jannach, D. (2018) Sequence-Aware Recommender Systems. ACM Computer Survey 1, 1, Article 1, 35 pages. https://doi.org/0000001.0000001

Soaresý F. Costaý D. and Dologý, P.(2019): Collective Embedding for Neural Context-Aware Recommender Systems. *In Thirteenth ACM Conference on Recommender Systems (RecSys '19), September 16–20, 2019, Copenhagen, Denmark.* https://doi.org/10.1145/3298689.ý3347028

Yifan Z. Tahir L. Sousa M, Vikas K(2018): A Field Study of Related Video Recommendations: Newest, Most Similar, or Most Relevant? *RecSys '18, October 2–7, 2018, Vancouver, BC, Canada.*

Twardowski, B.(2016): Modelling Contextual Information in Session-Aware Recommender Systems with Neural Networks. *RecSys '16, September 15-19, 2016, Boston , MA, USA*. DOI: http://dx.doi.org/10.1145/2959100.2959162