Hybrid Recommender Models in E-Learning: A Comprehensive Review of HybridBERT4Rec

Leon Knorr[1]
[1]Mannheim Master of Datascience

Student ID:1902854
8th January, 2024

Keywords: Education, AI, E-Learning

---

1 Introduction

In the dynamic landscape of today's knowledge-driven society and global economics, the concept of life-long learning has evolved into a cornerstone for personal and professional development. It ensures the competitiveness of individuals in a globalized market, while also aiding in overcoming social challenges, such as demographic change, social cohesion or public health through continuous education. Because of its key-role in overcoming today's challenges, many governments and corporations invested heavily in lifelong learning offerings and frameworks [1]. This investment can be seen by observing the vast landscape of different learning platforms, which have emerged over the last couple of years, ranging from publicly funded E-Learning solutions such as Moodle[2] or ILIAS[3], to private corporate platforms such as Linked-In Learning[4]. As individuals embark on continuous learning journeys on such platforms, the demand for tailored educational experiences has surged. These experiences not only include the recommendation of new content, but also content which aims to aid the user with current learning objectives. Recommender algorithms play a pivotal role in shaping these learning odysseys by providing personalized content recommendations [5].
In this work, the hybrid content recommendation system HybridBERT4Rec, which uses a transformer based approach to collaborative and content-based filtering techniques, is reviewed and adapted to an E-Learning use case.

2 Related Work

3 Sequential Content Recommendation

Traditional content recommendation systems usually observe user interactions as a set of independent and unrelated data points. These systems then compute or form a hidden representation in order to model user interests, which can then be used to predict which items may be relevant for the given user. If an item is considered relevant, it gets recommended. This modeling technique models a user's *general* preferences and interests [6]. But, this is insufficient modelling, as user preferences change over time [6]! For example, let Alice be a user whose general interests reflect romantic films, such as Titanic, Romeo & Juliet or "me before you" and who recently got into the Marvel Universe, and started watching Movies like "Spider-Man". In addition, Alice is now really interested in following up with that series. Alice's movie history contains a total of 10 movies, 9 romantic movies and one Spider-Man movie, with the latter being the most recently watched. A traditional content recommendation system would observe Alice's movie history and model her interests as consisting of 90% romantic films and 10% Marvel. As a result, the recommendation system would assign romantic films a higher relevance score than a movie that is similar to Spider-Man, leading to unsatisfying recommendations for Alice's *current* interests. Sequential Content Recommendation aims to solve this problem by observing user-interactions as a sequence. Thus, these models not only consider the items in the user history, but also the temporal aspect given by the order in which the items occur [6]. This ordering implies that more recent interactions are more relevant for recommending the next item, but at the same time also covers a user's general interest. As a result, Sequential Recommendation models are able to make accurate recommendations at any point in time, that also reflect temporary spikes of interests, while at the same time being able to recommend content that the user may generally be interested in. By choosing sequences as their data model, these models also gain a lot of flexibility in terms of the use cases they are able to cover. Because Sequential Recommendation models do not predict the relevance of items directly, but predict user-interaction probabilities at a given time-step, they are able to not only make recommendations for the present, but also for the future [6]. This allows these models to predict sequences of items, which can be used to craft recommendation series. E.g. a movie recommendation system can then recommend a complete series of movies which fit to Alice's interest in Marvel films, by predicting not only the next movie the user is likely to interact with but the next n movies. These reasons, render Sequential Recommendation models theoretically superior to traditional recommendation models in terms of content recommendation.

A common approach to realizing Sequential Recommendation models is by using Recurrent

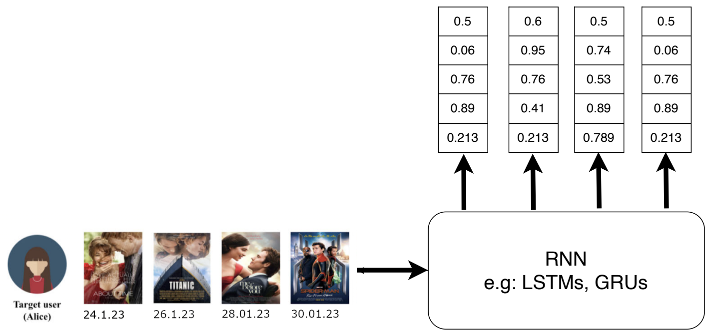

Fig. 1. Sequential Content Recommendation with RNNs [7]

Neural Networks (RNNs), such as Long-Short-Term-Memory (LSTMs) or Gated-Recurrent-Units (GRUs), as depicted in Figure 1. The RNN gets the sequence of user-interactions or items as input, in this case movies, and encodes them into a fixed length vector representation, which gets updated at every time-step in the sequence. The model then predicts the interaction probabilities or rating scores for every item which may be recommended at the next time step [7]. An example, for such an application can be found in Fung Yu et al. DREAM model [7]. However, these models suffer from common RNN problems, such as catastrophic forgetting, vanishing gradients and uni-directionality, as well as their inefficiency when dealing with longer sequences. This motivates the use of transformer based models, more specifically HybridBERT4Rec, which allows to use a sequence based approach to content recommendation without the downsides of RNNs [8].

4 HybridBERT4Rec

HybridBERT4Rec is a hybrid recommendation model, which was originally developed by Chanapa Channarong et al. [8] on a movie recommendation task [8]. It uses a combination of

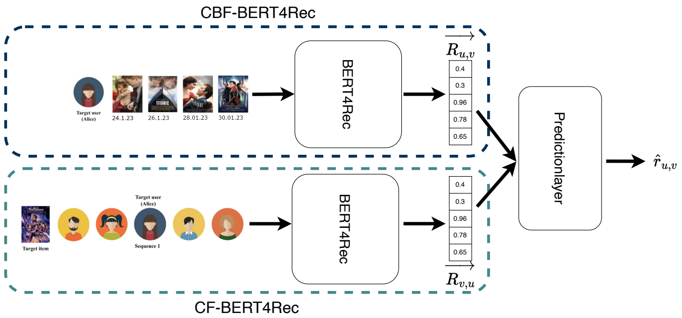

Fig. 2. High level overview of HybridBERT4Recs Architecture. [8]

Collaborative Filtering (CF) and Content Based Filtering (CBF) methods, in order to predict a relevance score for a given target item. As shown in Figure 2, it consists of three main parts:

1. CBF-BERT4Rec: Models Content Based Filtering
2. CF-BERT4Rec: Models Collaborative Filtering
3. Prediction layer: Combines both approaches and predicts the final relevance score $\hat{r}_{u,v}$

Both the CBF- and CF-Part of the model build upon HybridBERT4Recs predecessor BERT4Rec, and use the exact same architecture, while working with different inputs, in order to model their respective filtering techniques.

4.1 BERT4Rec

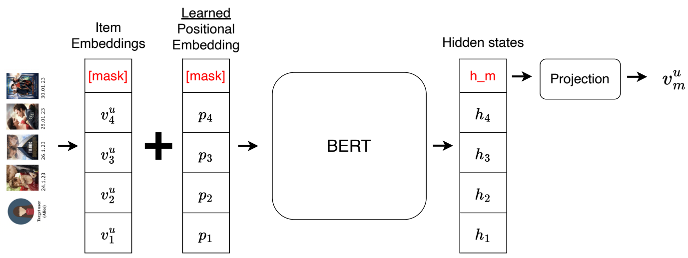

Fig. 3. BERT4Rec Architecture, taking item embeddings v_t^u from user u's history as input and predicts the next item v_m^u, u is likely to interact with [9].

BERT4Rec is a content recommendation model, which is based on the Bidirectional Encoder Representations from Transformers (BERT) model in order to facilitate Content Based Filtering. It has been developed in order to predict the next item a user is likely to interact with from a sequence of item embeddings from a user's interaction history. As depicted in Figure 3 The model consists of a learned positional embedding layer, an unmodified BERT transformer and a projection layer. In order to train BERT4Rec, Fei Sun et al. [9] applied Masked Language Modelling, a common training technique from Natural Language Processing and transferred it to the domain of recommendation models. As a result, during training, one or more items in the input sequence are masked at random and replaced with a special `[mask]` token. Then, the learned positional embeddings are added to the item embeddings, and the sequence gets passed on to the standard BERT transformer. The resulting hidden representation, of the masked item(s) is then passed through the prediction layer, which consists of a two layer feed forward neural network with GELU activations and a softmax layer, as given by equation 1, which produces an output distribution over the target items, the model is supposed to rank

$$P(v) = \mathrm{softmax}(\mathtt{GELU}(h_t^L W_p + b_p) E_T + b_O)[9] \tag{1}$$

h_t^L marks the hidden representation of the masked token, W_p is the weight matrix of the linear layer, b_p and b_O are bias terms and E_T is the item embedding matrix, which is also used to convert the items in the input sequence to item embeddings. The embedding matrix was reused in order to alleviate overfitting and reduce model size [9].
This training objective allows the transformer to learn strong and meaningful item representations from their contexts. In the domain of recommendation models, this translates to learning how to construct the item representation of the next optimal item a user would want to interact with from the user's history. The prediction layer then essentially calculates similarity scores between this optimal item representation and the target items. Despite its upsides, using BERT4Rec in a standalone manner, still comes with a limitation. As the prediction is solely based on extracting user historical patterns which can be seen as characteristics the user favors, the model can't recommend items from categories that aren't included in the user history already. In other words, it can't help a user discover new content, which is a general downside of using CBF in a standalone setting [8].

4.2 CF-HybridBERT4Rec

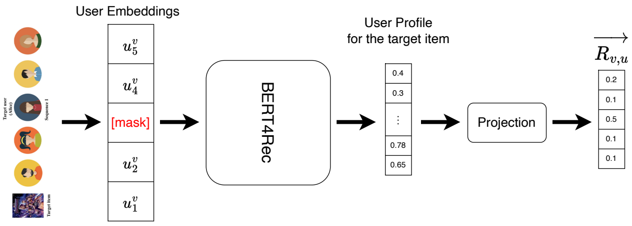

Fig. 4. CF-HybridBERT4Rec Architecture, taking user embeddings from all users that have rated the target item v as input and predicts the "target item profile $\overrightarrow{R_{v,u}}$" [8].

4.3 CBF-HybridBERT4Rec

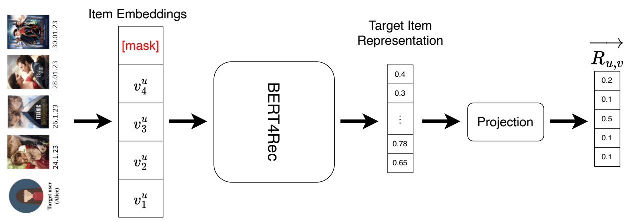

Fig. 5. CBF-HybridBERT4Rec Architecture, taking item embeddings from a user u history as input and predicts a "target user profile $\overrightarrow{R_{u,v}}$" [8].

4.4 Prediction Layer

$$\hat{r}_{u,v} = \sigma(WR(u,v) + b), \text{ with } R(u,v) = R_{uv} \odot R_{vu} \tag{2}$$

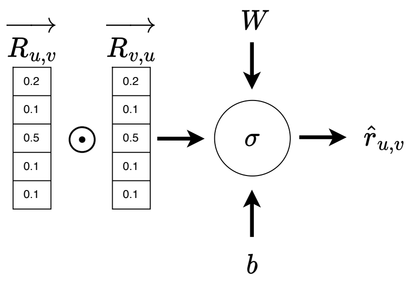

Fig. 6. Schematic of HybridBERT4Recs Prediction layer, which uses a generalization of Matrix Factorization based on Neural Networks with Sigmoid activations to predict the rating $\hat{r}_{u,v}$ user u would assign to item v [8].

4.5 Strengths & Weaknesses

4.6 Performance & Experiments

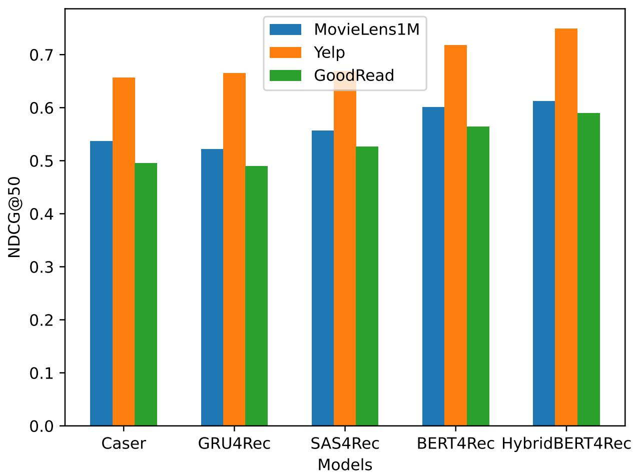

Fig. 7. Performance comparison of different recommender models on three datasets as published by the authors of HybridBERT4Rec [8].

5 Applying HybridBERT4Rec in an E-Learning Environment

5.1 The Trivial Solution

5.2 The Setting

5.3 Model Adaption

$$H(u) := (\{(x_i, t_j, s_k) | (x_i, t_j) \in R_{t,x}\}, \leq) \tag{3}$$

$$I(u) := (\{x_i | (x_i, t_j, s_k) \in H(u)\}, \leq) \tag{4}$$

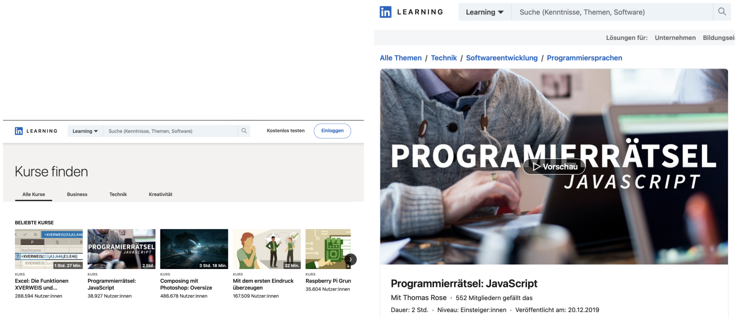

Fig. 8. Linked-In Learning landing-page and course overview [4].

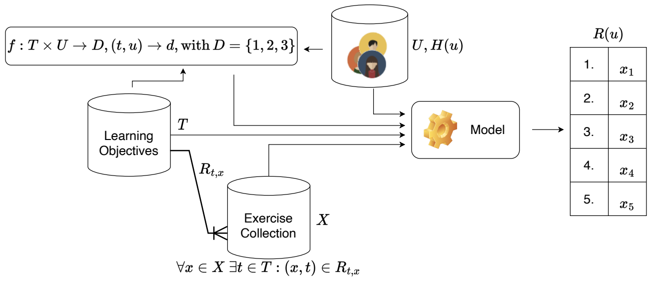

Fig. 9. The Setting, consisting of a user collection U and their histories $h(u)$, a collection of learning objectives T and a collection of exercises X, which can be used to predict a ranking $R(u)$ for a given user u.

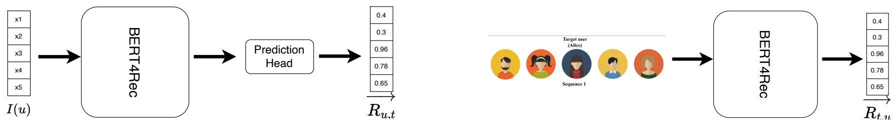

Fig. 10. The adapted CBF-HybridBERT4Rec model on the left and the adapted CF-HybridBERT4Rec model on the right.

$$u \in N \Longleftrightarrow d_{u,t} = d_{u_m,t} \land (x,t) \in \{(x,t) | (x,t,s_k) \in H(u)\} \tag{5}$$

Algorithm 1 HybridBERT4Rec in an E-Learning Setting

for all $u_m \in U$ **do**
  $r_{x,u_m} =$ `cbf_hybridbert4rec`$(H(u_m))$
  for all $(x,t) \in R_{t,x}$ **do**
    $r_{u,x} =$ `cf_bert4rec`(u_m, t, x)
    $\hat{r}_{u,x} =$ `prediction_layer`$(r_{x,u}, r_{u,x})$
  end for
end for

5.4 Solving Evaluation

ABC

5.5 Remaining Possible Issues

ABC

6 Conclusion

ABC

7 References

1. Rubenson, K. *Adult Learning and Education* (Academic Press, Feb. 2011).
2. *Startseite — Moodle.org* https://moodle.org/.
3. *Ilias.De* https://www.ilias.de/.
4. *LinkedIn Learning mit Lynda: Onlinekurse aus dem Business-, Technik- und Kreativbereich* https://de.linkedin.com/learning/.
5. Jeevamol, J. & Renumol, V. G. An Ontology-Based Hybrid e-Learning Content Recommender System for Alleviating the Cold-Start Problem. *Educ Inf Technol* **26,** 4993–5022 (July 2021).
6. Wang, S., Hu, L., Wang, Y., Cao, L., Sheng, Q. Z. & Orgun, M. Sequential Recommender Systems: Challenges, Progress and Prospects, 6332–6338 (2019).
7. Yu, F., Liu, Q., Wu, S., Wang, L. & Tan, T. *A Dynamic Recurrent Model for Next Basket Recommendation* in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Association for Computing Machinery, New York, NY, USA, July 2016), 729–732.
8. Channarong, C., Paosirikul, C., Maneeroj, S. & Takasu, A. HybridBERT4Rec: A Hybrid (Content-Based Filtering and Collaborative Filtering) Recommender System Based on BERT. *IEEE Access* **10,** 56193–56206 (2022).
9. Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W. & Jiang, P. *BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer* Aug. 2019. arXiv: `1904.06690 [cs]`.