

A Comprehensive Exploration of Item Recommendation Text Ranking

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

28-07-2022 / 29-07-2022

CITATION

Guo, Tong (2022): A Comprehensive Exploration of Item Recommendation Text Ranking. TechRxiv. Preprint.
<https://doi.org/10.36227/techrxiv.20388159.v1>

DOI

[10.36227/techrxiv.20388159.v1](https://doi.org/10.36227/techrxiv.20388159.v1)

A Comprehensive Exploration of Item Recommendation Text Ranking

Tong Guo

779222056@qq.com

Abstract. In this paper, we study the personalized ranking problem for item’s recommendation text. Item’s recommendation text is the sentence that describe the item highlights for user decision (such as buy or click). The recommendation texts is shown under the item, and we also call them rec-texts. One item has multiple rec-texts, and different rec-texts have different affect for user decision. So the problem is to capture the user preference for each item’s rec-texts and to personalized display. In this paper, we study multiple methods to train a model which learn the user preference scores for each item’s rec-texts. The online experimental results demonstrate our methods work.

Keywords: Deep Learning · Recommendation System

1 Introduction

Recommendation systems intent to predict each user’s preference on candidate items. Our problem is similar to this user-item interest match problem. In this paper, each item has multiple sentences that describe the item highlight, we also call them rec-texts. We study to display the rec-texts that meet the user’s preference. For example, we display the rec-texts that describe the kind of items for the users who like the same kind of items. The item can only display 1-3 rec-texts and the candidate rec-texts number is 10-30.

The training data is from the users’ online log of clicks and orders. The main hard part for this problem is: The users click or buy the item is not because the users like the rec-texts of the item, but because the users like the item itself. In order to solve this problem, we view this problem as a three-part matching problem. In click through rate (CTR) prediction problem we train a model from the user-item matching data. In this personalized rec-texts prediction problem, we train a model that learns from the user-item-text triples.

In summary, our contributions in this paper are as follows:

To the best of our knowledge, this is the first work to study the personalized ranking problem for item’s rec-texts.

We study multiple methods to solve this problem.

2 Related Works

Our problem is similar to the CTR problem which can be viewed as a user-item matching task. In this view, our problem can be view as a user-item-text

matching task or user-text matching task. So there many models we can refer to, such as Deepfm[1] and AutoInt[3].

3 Problem Definition

Given a user and a list of items, the task is to rank the candidate rec-texts for each item. The target is to improve the revenue per thousand impressions (RPM). The training dataset is from the click and order log. We view this problem as a click through rate (CTR) prediction problem to solve it. The learning target can be click or order for each user-item-text triple input of features.

4 Our Methods

In this section we describe the features we use first. Then we describe the deep model we use. In the end of this section we describe learning target/loss.

4.1 The Input Features

Based on the task, we design three parts of features: the user features, the item features, the text tokens embedding as feature. The user features contain user age, user item click history, user item order history, user's rec-text click history. The item features contain item types, item average CTR. We also remove the item features for the model because it is a better text choosing problem in each item.

4.2 The Deep Model

We use deepfm[1] added transformer[2] as our model. Each feature is processed into id embedding and feed to the deep model. The model's output is a 0-1 score which represents a user's preference for each item's rec-text. The word embedding is randomly initialised. The detail is shown in Fig 1 and Fig 2. The code is refer to [4] and [5].

4.3 The Learning Target

We use CTR-based loss and RPM-based loss. For the CTR-based loss, the learning target is the click-view-rate for each user-item-text in the past days. For the CTR-based loss, the learning target is the order-view-rate for each user-item-text in the past days.

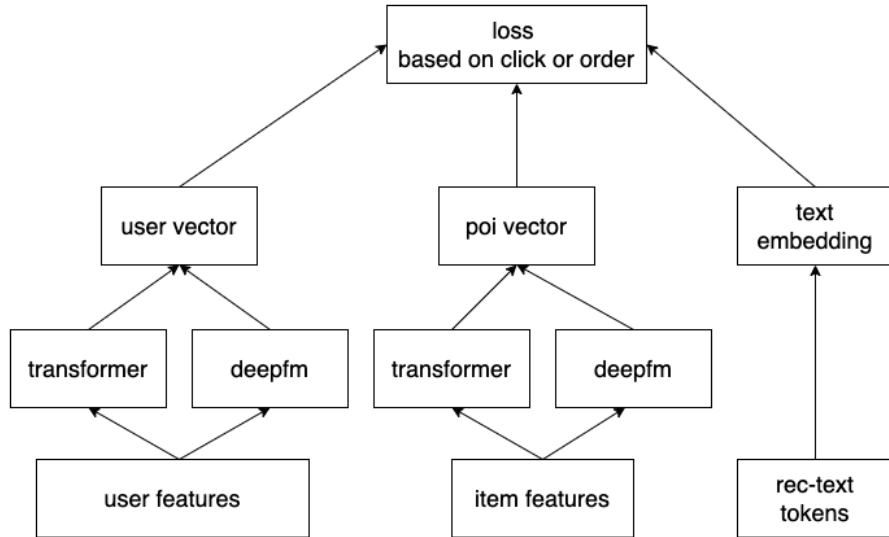


Fig. 1. The overall deep model framework, if we view the problem as user-item-text matching task.

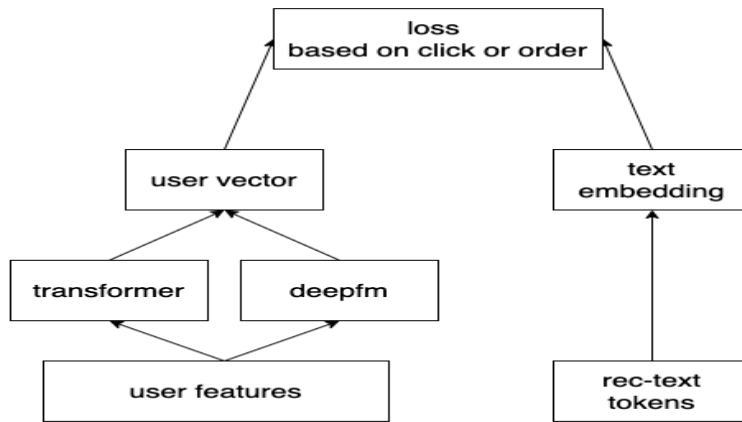


Fig. 2. The overall deep model framework, if we view the problem as user-text matching task.

Methods	Target/Loss	offline AUC
user features + item features + text embedding	CTR-based loss	0.77
user features + text embedding	CTR-based loss	0.69
user features + item features	CTR-based loss	0.74
user features + item features + text embedding	RPM-based loss	0.88
user features + text embedding	RPM-based loss	0.78
user features + item features	RPM-based loss	0.85

Table 1. The offline AUC results.

Methods	Target/Loss	online CTR	online RPM
user features + item features + text embedding	CTR-based loss	+0.0%	+0.10%
user features + text embedding	CTR-based loss	+0.0%	+0.15%
user features + item features + text embedding	RPM-based loss	+0.0%	+0.13%
user features + text embedding	RPM-based loss	+0.0%	+0.17%

Table 2. The online CTR and RPM results.

5 Results and Analysis

The offline AUC and online experimental results is shown in table 1 and 2. The baseline for online results (Table 2) is random rec-texts ranking for each item. The table 3 shows the results of ablation study for deep model.

The experimental results show that the without-item-feature is better than the full features setting. In fact it is a best-text choosing problem under each item. The candidate texts of a item is fixed for all user. So the conclusion is that the item features is useless.

We train the model by 0/1 classification target and 0-1 regression target, we have not found much more difference. Because we think it is hard to capture the detail gain of rec-text for the 0-1 regression target. As whether the user click or order the item is only slightly affected by the rec-texts of the item.

The reason for online CTR that is not increasing, is because user’s click-number do not increase. The rec-text can help user to make decision. User only order more and do not click more on the item list.

6 Conclusion

In the paper, we solve the personalized ranking problem of each item’s candidate rec-texts. We view the problem as a CTR problem. The best feature setting in our scene is using user features and rec-text feature/embedding. As we get conclusion that the item features is useless. The best loss setting is using the order/RPM-based label as the training target. The online results shows that our method can increase the performance of the item list by personalized ranking the rec-texts of each item.

Models	Target/Loss	offline AUC
deepfm	CTR-based loss	0.75
deepfm + transformer	CTR-based loss	0.77
deepfm	RPM-based loss	0.86
deepfm + transformer	RPM-based loss	0.88

Table 3. The results of ablation study for deep model, with item features, user features and text embedding.

References

1. Guo H, Tang R, Ye Y, et al. DeepFM: a factorization-machine based neural network for CTR prediction[J]. arXiv preprint arXiv:1703.04247, 2017.
2. Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.
3. Song W, Shi C, Xiao Z, et al. AutoInt: Automatic feature interaction learning via self-attentive neural networks[C]//Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 2019: 1161-1170.
4. github.com/quotong1988/movielens_dataset
5. github.com/quotong1988/criteo_dataset