Boosting Deep CTR Prediction with a Plug-and-Play Pre-trainer for News Recommendation

Qijiong Liu
The Hong Kong Polytechnic University
jyonn.liu@connect.polyu.hk

Jieming Zhu
Huawei Noah's Ark Lab
jiemingzhu@ieee.org

Quanyu Dai
Huawei Noah's Ark Lab
quanyu.dai@connect.polyu.hk

Xiaoming Wu[*]
The Hong Kong Polytechnic University
xiao-ming.wu@polyu.edu.hk

Abstract

Understanding textual content is critical to improving the quality of news recommendation. To achieve this goal, recent studies have proposed to apply pre-trained language models (PLMs) such as BERT for semantic-enhanced news recommendation. Despite their great success in offline evaluation, it is challenging to apply such large PLMs in real-time ranking tasks due to the stringent latency requirements in model updating and inference. To bridge this gap, we propose a plug-and-play pre-trainer, namely PREC, to learn both user and news encoders through multi-task pre-training. Instead of directly leveraging sophisticated PLMs for end-to-end inference, we focus on how to use the cached user and item representations to boost the performance of traditional ID-based models for click-through-rate prediction. This enables efficient online inference as well as compatibility to the widely-used models in industry, which would significantly ease the practical deployment. We validate the effectiveness of PREC through both offline evaluation on public datasets and online A/B testing in an industrial system.

1 Introduction

Personalized news recommendation has become a ubiquitous channel in various online applications, such as Google News and MSN News, which helps users discover their interested news information. To deal with the massive amounts (usually millions) of daily news, industrial recommender systems usually apply a multi-stage recommendation pipeline as illustrated in Figure 1. It generally involves two phases, matching and ranking (Covington et al., 2016). The matching phase first generates hundreds or thousands of news candidates from multiple channels (e.g., popularity-based channel, content-based channel (Wu et al., 2019a,b), and collaborative filtering channel (Linden et al., 2003; Sedhain et al., 2014)). Subsequently, the ranking phase employs a news ranking task, such as click-through rate (CTR) prediction, to rank the candidate news and finally return the top dozens of news to the user interface.

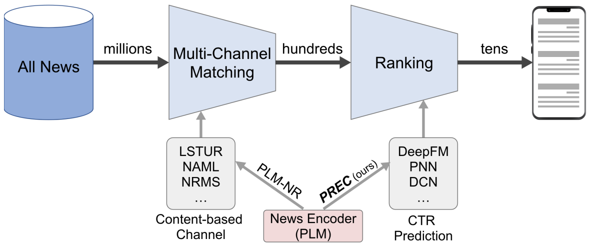


Figure 1: A multi-stage pipeline for news recommendation.

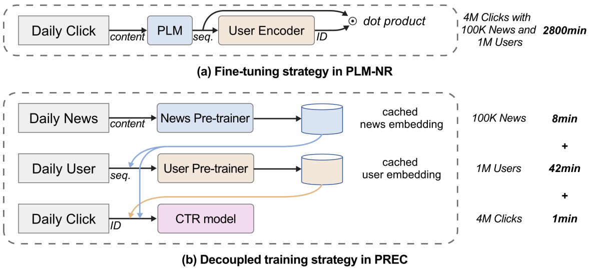
The matching and ranking phases have different goals and requirements. Concretely, the matching phase aims for efficient and high-recall candidate retrieval from millions of news corpus. Therefore, most studies construct two-tower networks (e.g., NPA (An et al., 2019), LSTUR (Wu et al., 2019a), and NRMS (Wu et al., 2019b)) to learn user representations and item representations separately, and then apply simple dot product to compute similarity scores. As such, both user and item representations can be pre-computed offline and cached online for fast nearest neighbour search (e.g., using Faiss (Johnson et al., 2019)). In contrast, the ranking phase targets at accurate scoring of each candidate item based on CTR prediction and thus requires networks to capture complex feature interactions between users and news (e.g., DeepFM (Guo et al., 2018), PNN (Qu et al., 2016), and DCN (Wang et al., 2017)). These widely adopted CTR prediction models in industry are usually small (e.g., 3 $\sim$ 5 layers) to meet the latency requirements for online inference and enable frequent model updates (e.g., hourly or daily).

To leverage powerful pre-trained language models (PLMs) to better capture the semantics of news

[*] Corresponding author.

Proceedings of the 29th International Conference on Computational Linguistics, pages 2823–2833
October 12–17, 2022.

content, some recent studies (Zhang et al., 2021a; Wu et al., 2021b,a) propose the use of PLMs for news recommendation. For example, as depicted in Figure 1, PLM-NR (Wu et al., 2021a) replaces original news encoders (e.g., CNN in Krizhevsky et al. (2012) and multi-head attention in Vaswani et al. (2017)) with PLMs such as BERT (Devlin et al., 2019) to empower existing content-based matching models (e.g., LSTUR (An et al., 2019), NAML (Wu et al., 2019a), and NRMS (Wu et al., 2019b)). Yet, these studies typically adopt the common pretrain-finetune strategy to train PLM-based news encoder and user encoder jointly, which is extremely time-consuming given the massive amount of click data. As a concrete example illustrated in Figure 2, we estimate the updating time of the end-to-end model and the hierarchically decoupled model on MIND (Wu et al., 2020). Finetuning "PLM-NR + NRMS" with the MIND click data (4M clicks with 1M users and 100K news) takes 2800 minutes for one epoch. This might be tolerable during matching as reported in Wu et al. (2021a), where user and item representations are pre-computed offline, but is absolutely unacceptable for CTR prediction tasks.



(a) Fine-tuning strategy in PLM-NR

(b) Decoupled training strategy in PREC

Figure 2: Model updating strategy of PLM-NR-like model (left) and our PREC-like model (right) on the MIND dataset. The "content", "seq", and "ID" symbols respectively denote news content, user browsing sequence, and ID-based features, which are fed to the following model. Please refer to subsubsection 4.1.2 for detailed settings.

In this paper, we explore the use of PLMs for CTR prediction tasks and propose a plug-and-play pre-trainer, namely PREC, to learn both user and news encoders through multi-task pre-training. In particular, our PREC model has the following key advantages: 1) This is the first work to integrate both news pre-trainer and user pre-trainer for recommendation. The former fuses multi-view news features (e.g., title, abstract, category) for representation learning, while the latter learns user representations from historical interaction sequences depending on the learned news representations and iter-dependencies among them. 2) The PREC model is constructed in a hierarchical decoupled manner, bringing rich news contents and deep user interests to CTR models while meeting the requirements of both inference and updating efficiency. As shown in Figure 2, PREC's model updating time only reaches about 50 minutes, which is 50+ times faster than PLM-NR under the same setting. 3) The decoupled design makes PREC easily compatible with various existing CTR prediction models (e.g., PNN (Qu et al., 2016), DCN (Wang et al., 2017), DeepFM (Guo et al., 2018)), where user and news representations learned from PREC could be used as features or embedding initialization for downstream models. This allows PREC to be easily deployed in industrial recommender systems.

We conduct extensive experiments to validate the effectiveness of PREC on two open benchmark datasets for news recommendation, MIND (Wu et al., 2020) and Adressa (Gulla et al., 2017). The experiments show that our pre-trainer PREC can consistently boost the performance of existing lightweight CTR prediction models, even outperforming some state-of-the-art heavyweight models. In addition, we have deployed PREC in the production system and an online A/B test shows that PREC leads to a 2.4% improvement on the overall CTR metric.

2 Related Work

News recommendation has become increasingly popular in recent years. Models based on deep neural networks (Wang et al., 2018; Wu et al., 2019b) have been proposed to learn news and user representations by leveraging CNN (Krizhevsky et al., 2012) or attention mechanism (Vaswani et al., 2017). These end-to-end models focus on solving recommendation task and typically have limited ability in semantic understanding. Most recently, PLMs such as BERT (Devlin et al., 2019) have shown promising results in news recommendation (Zhang et al., 2021a; Wu et al., 2021c; Li et al., 2022). Some works (Wu et al., 2021a,b; Yu et al., 2021) leverage PLMs to learn news embeddings by exploring out-domain knowledge and learning news semantics. These models can be deployed in matching stage, but not compatible to the ranking stage due to intolerant model updating time. Some use PLMs to learn user representations (Wu et al., 2021c; Sun et al., 2019; Chen et al., 2019) by modeling user interests with historical news sequences. Besides, we notice that SpeedyFeed (Xiao et al., 2022) is presented to semi-decouple PLM-based recommender models for faster training, but the model updating time is still too long.

While pre-trained models have shown great promise in content understanding, lightweight CTR models (Wang et al., 2017; Guo et al., 2018; Lian et al., 2018) still dominate industrial applications due to its high efficiency, where PLMs cannot be deployed due to its large size. Our proposed pre-trainer can work seamlessly with these industrial CTR models and provide them semantic news embeddings and user interest knowledge with almost no overhead in inference time and deployment cost.

Model Pre-Training. The pioneer works (Mikolov et al., 2013; Pennington et al., 2014) introduced pre-training for natural language understanding by providing a static representation for each word. Later, some methods (Peters et al., 2018; Devlin et al., 2019) proposed to dynamically generate word representations according to the context, which effectively resolves the ambiguity problem. Very recently, the huge success of BERT has led to a surge of interest in pre-training models. A line of works (Zhou et al., 2020; Xie et al., 2020; Wu et al., 2021a,c) have proposed to use pre-training for new recommendation. However, the pre-training tasks for news recommendation are under-explored. In this paper, we propose multiple tasks for multi-view news and user pre-training.

3 The Proposed Method

In this section, we present our proposed cascaded pre-trainer for news recommendation (PREC). We first introduce the model framework, pre-training tasks and pre-training strategy. Then, we describe how to apply the pre-trainer for downstream tasks.

3.1 Model Framework

Figure 3 presents the model framework of our proposed cascaded pre-trainer PREC, which consists of a news pre-trainer and a user pre-trainer. Both news and users contain multiple feature sets, i.e., multiple views. For example, a news has title, entities and abstract, and a user is usually characterized by her/his historical browsed news, location, personal information, and so on. To capture the rich semantics from these multi-view data, PREC is designed to capture both intra-view and inter-view knowledge based on Transformer, with the news pre-trainer aimed at learning the deep semantic meanings of the textual content while the user pre-trainer targeting at capturing the inherent user interests. Since browsing history is one of the most significant views of users, the user pre-trainer relies on the news embeddings learned from the news pre-trainer. Thus, our model is built and trained in a cascaded manner.

3.1.1 News Pre-trainer

We assume each news contains n views (e.g., title, entities, abstract) and each view contains a sequence of tokens. We use $\boldsymbol{t}_i = \left[t_{i,1}, ..., t_{i,|\boldsymbol{t}_i|}\right]$ to represent the token sequence of the i-th view, where $i \in \{1, ..., n\}$ and $|\boldsymbol{t}_i|$ is the total length of the view. We propose a news pre-trainer to comprehend the deep semantic meanings of news by taking a concatenated sequence of the multi-view content as input. Following BERT (Devlin et al., 2019), the input sequence starts with a special token $\langle$CLS$\rangle$, and views are separated by another special token $\langle$SEP$\rangle$. For simplicity, these two special tokens are omitted in the following description. After concatenating n views, the input token sequence is represented as $\boldsymbol{t} = \left[t_{1,1}, ..., t_{1,|t_1|}, ..., t_{n,1}, ..., t_{n,|t_n|}\right]$, and the token embedding can be obtained through an embedding layer as follows:

$$\boldsymbol{E}^t_{token} = [\boldsymbol{e}^t_1, \boldsymbol{e}^t_2, ..., \boldsymbol{e}^t_s] \in \mathbb{R}^{s\times d}, \qquad (1)$$

where $s = \sum_i^n |\boldsymbol{t}_i|$ is the sequence length and d is the embedding dimension. We also use position embedding to encode the positional information of a token, and view embedding to distinguish between different views. We denote the position embedding and view embedding of a news as $\boldsymbol{E}^t_{pos}$ and $\boldsymbol{E}^t_{view}$, respectively. For a given token, the input representation is obtained by summing up the corresponding token, position embedding and view embedding. Then, given a news, the integrated input representation of the news pre-trainer is obtained as follows:

$$\boldsymbol{E}^t = \boldsymbol{E}^t_{token} + \boldsymbol{E}^t_{pos} + \boldsymbol{E}^t_{view}. \qquad (2)$$

The news pre-trainer consists of two submodules, news transformer and news aggregator. The news transformer first learns the refined token representations $\bar{\boldsymbol{E}}^t$ of news content through multiple Transformer layers, and then the news aggregator combines the token representations into a unified news representation $\bar{\boldsymbol{t}}$. The learned news

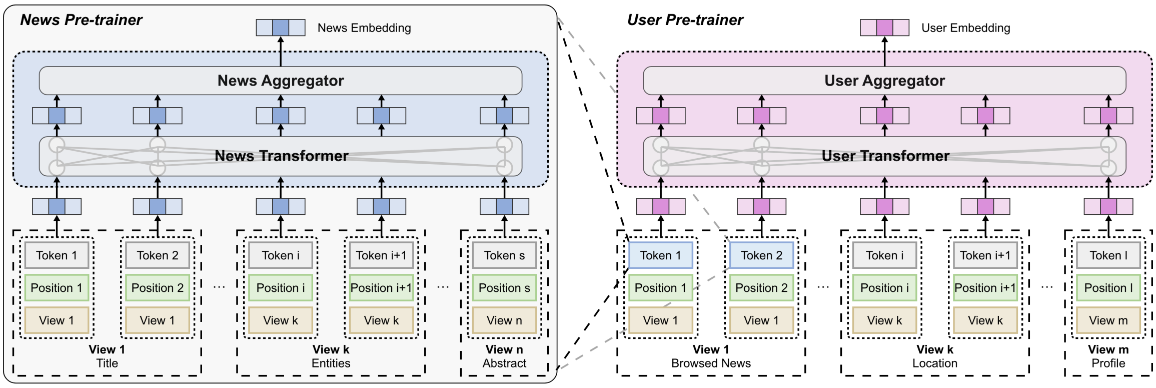


Figure 3: The architecture of our PREC model.

representation will be used for news token embedding in the user pre-trainer and the training of downstream CTR models.

3.1.2 User Pre-trainer

Similarly, users are characterized by m views, e.g., historical browsed news, location, and user profile. We use $\boldsymbol{u}_i = \left[u_{i,1}, ..., u_{i,|\boldsymbol{u}_i|}\right]$ to represent the i-th view of a user, where $i \in \{1, ..., m\}$, and $|\boldsymbol{u}_i|$ is the total length of the view. User pre-trainer has a similar model framework as the news pre-trainer, which consists of an embedding layer, multiple Transformer layers, and a aggregator. It takes a sequence of concatenated multi-view user information as input, which is represented as $\boldsymbol{u} = [u_{1,1}, ..., u_{1,|u_1|}, ..., u_{m,1}, ..., u_{m,|u_m|}]$. Firstly, the input token sequence of a user is transformed into three embedding sequences, including token, position, and view sequences, through an embedding layer. Specifically, the token embedding sequence is represented as follows:

$$\boldsymbol{E}^u_{token} = [\boldsymbol{e}^u_1, \boldsymbol{e}^u_2, ..., \boldsymbol{e}^u_l] \in \mathbb{R}^{l\times d}, \qquad (3)$$

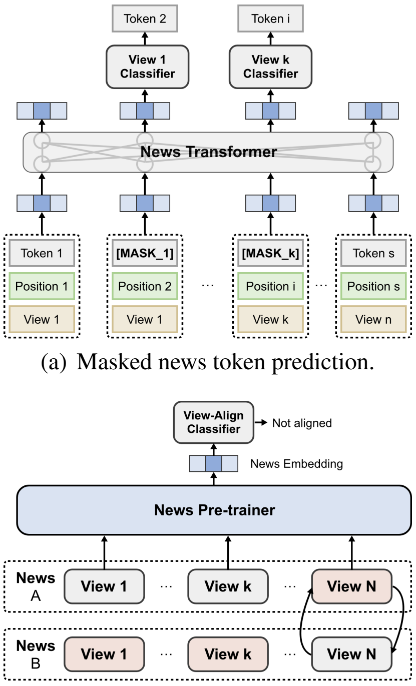
where $l = \sum_i^m |\boldsymbol{u}_i|$ is the sequence length. Similarly, we denote the corresponding position embedding and view embedding of a user as $\boldsymbol{E}^u_{pos}$ and $\boldsymbol{E}^u_{view}$, respectively. Then, the integrated input user representation for the user pre-trainer is obtained by summing the three types of embeddings:

$$\boldsymbol{E}^u = \boldsymbol{E}^u_{token} + \boldsymbol{E}^u_{pos} + \boldsymbol{E}^u_{view}. \qquad (4)$$

As demonstrated by existing work (An et al., 2019; Wu et al., 2019b), users' historical browsed news is key to discover user interests. For training efficiency, we initialize the token embeddings of user browsed news with the learned news embeddings from the news pre-trainer, and keep them fixed during training.

Then, the user transformer explores users' potential interests through multiple Transformer layers and produces the refined token representations $\bar{\boldsymbol{E}}^u$, and the user aggregator fuses the refined representations into a single user representation $\bar{\boldsymbol{u}}$. Note that we use average pooling as our news and user aggregators as in Zhang et al. (2021a).

3.2 Pre-training Tasks



(a) Masked news token prediction.

(b) News view alignment.

Figure 4: Two pre-training tasks in PREC.

Masked News Token Prediction (MNTP) is designed to learn context-aware token representations. The framework is shown in Figure 4(a). In this task, we first randomly sample a portion (15% in our experiments) of tokens in each sequence $\boldsymbol{t}$ and replace them with other tokens. Specifically, for each sampled token, there is an 80% probability of being replaced with a unique identifier ⟨`MASK_i`⟩, a 10% possibility of being substituted with a random token from the vocabulary, and another 10% probability of remaining unchanged. We then conduct a classification task on the sampled tokens with a multi-layer perceptron classifier by taking the corresponding refined token embedding from the news transformer as input. Thus, the MNTP task is to minimize the following negative log-likelihood loss:

$$L_{mntp}(\boldsymbol{\theta}, \boldsymbol{\beta}_1) = -\mathbb{E}_{\boldsymbol{t} \sim D^t} \log P(t_{i,j} | \boldsymbol{t}_{\backslash(i,j)}, \boldsymbol{\theta}, \boldsymbol{\beta}_1), \tag{5}$$

where $\boldsymbol{\theta}$ denotes the model parameters of the news transformer and aggregator, $\boldsymbol{\beta}_1$ denotes the model parameters of the classifier, D^t is the training set of the news pre-trainer, and $P(t_{i,j}|\boldsymbol{t}_{\backslash(i,j)}; \boldsymbol{\theta}, \boldsymbol{\beta}_1)$ is the probability of correctly predicting the current token $t_{i,j}$ given other tokens $\boldsymbol{t}_{\backslash(i,j)}$.

News View Alignment (NVA) aims to capture inter-view relations. The framework is presented in Figure 4(b). In this task, we manually construct out-of-alignment news by replacing some views of the considered news with those of a randomly sampled one, and treat them as negative samples. More precisely, for a piece of news, it will undergo one of the following two operations randomly: (a) remains unchanged (labeled as 1); (b) constructed as out-of-alignment news (labeled as 0). Then, the processed token sequence $\boldsymbol{t}'$ is fed into the news pre-trainer to generate the unified news representation $\bar{\boldsymbol{t}}$. We perform a binary classification task with $\bar{\boldsymbol{t}}$ as input by optimizing the following cross-entropy loss:

$$L_{nva}(\boldsymbol{\theta}, \boldsymbol{\beta}_2) = -\mathbb{E}_{\boldsymbol{t} \sim D^t} \log P(l | \boldsymbol{t}', \boldsymbol{\theta}, \boldsymbol{\beta}_2), \tag{6}$$

where $\boldsymbol{\beta}_2$ denotes the model parameters of the binary classifier, $l \in \{0, 1\}$ represents the types of news samples, and $\boldsymbol{t}'$ is the reconstructed news after the random operations.

News Category Prediction (NCP) is proposed to capture global knowledge of news. The extracted news embedding $\bar{\boldsymbol{t}}$ is leveraged to predict the category of the considered news. The loss function of this task is as follows:

$$L_{ncp}(\boldsymbol{\theta}, \boldsymbol{\beta}_3) = -\mathbb{E}_{\boldsymbol{t} \sim D^t} \log P(c | \boldsymbol{t}, \boldsymbol{\theta}, \boldsymbol{\beta}_3), \tag{7}$$

where $\boldsymbol{\beta}_3$ denotes the model parameters of the category classifier, and c is the news category.

We also design **Masked User Token Prediction (MUTP)** task and **User View Alignment (UVA)** task for the user pre-trainer, which are similar to the MNTP task and the NVA task, respectively. The loss functions of the MUTP task and the UVA task are as follows:

$$L_{mutp}(\boldsymbol{\phi}) = -\mathbb{E}_{\boldsymbol{u} \sim D^u} \log P(u_{i,j} | \boldsymbol{u}_{\backslash(i,j)}, \boldsymbol{\phi}, \boldsymbol{\varphi}_1), \tag{8}$$

$$L_{uva}(\boldsymbol{\phi}) = -\mathbb{E}_{\boldsymbol{u} \sim D^u} \log P(l | \boldsymbol{u}', \boldsymbol{\phi}, \boldsymbol{\varphi}_2), \tag{9}$$

where $\boldsymbol{\phi}$ denotes the model parameters of the user transformer and aggregator, $\boldsymbol{\varphi}_1$ and $\boldsymbol{\varphi}_2$ denote the model parameters of the classifier for the MUTP task and the NUA task respectively, D^u is the training set of the user pre-trainer, (i, j) is the index of the masked token, and $l \in \{0, 1\}$ represents whether the views of the reconstructed token sequence $\boldsymbol{u}'$ are aligned or not.

3.3 Pre-training Strategy

Stage One. The performance of the user pre-trainer heavily relies on the quality of news embeddings, thus we first conduct pre-training of the news pre-trainer to obtain informative news embeddings. In this stage, MNTP, NVA, and NCP tasks are performed to capture intra-view and inter-view semantic meanings of news, and the overall loss is formulated as:

$$L_{NP} = \omega_{mntp} L_{mntp} + \omega_{nva} L_{nva} + \omega_{ncp} L_{ncp}, \tag{10}$$

where ω_{mntp}, ω_{nva}, and ω_{ncp} are hyper-parameters for balancing the losses of different tasks.

Stage Two. After obtaining news embeddings from the news pre-trainer, we perform the MUTP and UVA tasks to optimize the user pre-trainer with the following loss:

$$L_{UP} = \omega_{mutp} L_{mutp} + \omega_{uva} L_{uva}, \tag{11}$$

where ω_{mutp} and ω_{uva} are hyper-parameters for balancing the losses of the two tasks.

This two-stage pre-training strategy has the following advantages. 1) It greatly simplifies the optimization of PREC by decoupling the optimization of the news pre-trainer and user pre-trainer. 2) It allows to update the two pre-trainers in an asynchronous manner, which nicely satisfies the requirements of real industrial scenarios where user representations are likely to be updated more frequently due to the constant change of user interests.

Table 1: Dataset statistics after preprocessing.

Number	MIND		Adressa	
	small	large	1Week	4Week
#News	65,238	104,151	81,018	81,018
#Users	94,057	750,434	214,464	400,279
#Clicks	347,727	3,958,501	354,181	877,605
#Samples	8,584,442	97,592,931	1,770,905	4,388,025

Table 2: Comparison results between baseline models and PREC-boosted models on the MIND-small and Adressa-1Week datasets.

MIND-small				
Method	DCN	DCN$_{+NP}$	DCN$_{+NP+UP}$	Improv.
AUC	65.07	66.63	**67.47**	3.69%
MRR	33.12	33.62	**34.88**	5.31%
nDCG@5	34.02	34.95	**36.20**	6.41%
nDCG@10	40.06	41.23	**42.33**	5.67%
Method	PNN	PNN$_{+NP}$	PNN$_{+NP+UP}$	Improv.
AUC	61.80	65.24	**66.55**	7.69%
MRR	30.11	32.42	**33.18**	10.20%
nDCG@5	30.65	30.02	**34.86**	13.74%
nDCG@10	36.93	40.14	**40.89**	10.72%
Method	DeepFM	DeepFM$_{+NP}$	DeepFM$_{+NP+UP}$	Improv.
AUC	61.86	65.46	**65.98**	6.66%
MRR	29.77	33.26	**33.42**	12.26%
nDCG@5	30.15	34.31	**34.92**	15.82%
nDCG@10	36.74	40.73	**40.94**	11.43%
Adressa-1Week				
Method	DCN	DCN$_{+NP}$	DCN$_{+NP+UP}$	Improv.
AUC	75.86	82.39	**83.78**	10.44%
MRR	93.26	95.07	**95.61**	2.52%
nDCG@5	95.30	96.60	**97.31**	2.11%
nDCG@10	95.51	96.81	**97.32**	1.90%
Method	PNN	PNN$_{+NP}$	PNN$_{+NP+UP}$	Improv.
AUC	71.21	79.64	**81.92**	15.04%
MRR	92.05	94.06	**95.09**	3.30%
nDCG@5	94.67	95.14	**96.80**	2.25%
nDCG@10	94.83	95.74	**96.89**	2.17%
Method	DeepFM	DeepFM$_{+NP}$	DeepFM$_{+NP+UP}$	Improv.
AUC	73.40	79.64	**81.14**	10.54%
MRR	92.43	94.29	**94.68**	2.43%
nDCG@5	94.43	95.99	**96.07**	1.74%
nDCG@10	94.83	96.22	**96.42**	1.68%

3.4 CTR Prediction with PREC

The learned news and user representations from PREC are leveraged to boost the performance of deep CTR prediction models. Generally, deep CTR models, such as PNN (Qu et al., 2016), DeepFM (Guo et al., 2018), and DCN (Wang et al., 2017), take multiple user-side (e.g., *user id*, *browsed news ids*, *browsed news title sequence*) and news-side (e.g., *news id*, *title*) features as input. These raw features are usually first transformed into multi-field categorical format through a feature engineering module before being fed to a CTR model. Then, the input features are transformed into continuous embedding vectors through an embedding layer of the model (Zhang et al., 2021b). We use the pre-trained representations to initialize the embedding vectors of news id and user id to enrich the feature inputs of CTR models. Note that we keep the pre-trained news and user representations fixed during model training and add additional transformation matrices to project them into the same space of the embedding vectors of other features. The two transformation matrices are part of trainable model parameters and updated during optimization.

4 Experiments

4.1 Experimental Settings

We conduct experiments on two large real-world news recommendation datasets: MIND (including small and large versions) (Wu et al., 2020) and Adressa (including the 1-week and 4-week versions) (Gulla et al., 2017). Our experiments include two phases: pre-training and plug-and-play. For the MIND dataset, we use news title and abstract as news views, news category as the label in the NCP task, and user history as user view. As users' profiles are missing, we omit the UVA task on MIND. We set $\omega_{mntp} = \omega_{nva} = \omega_{ncp} = \omega_{mutp} = 1, \omega_{uva} = 0$. For the Adressa dataset, we take the news title, description, and keywords as news view and use user history, location (country, region, and city), and device information as user view. Since news category information is missing, we omit the NCP task on Adressa. We set $\omega_{mntp} = \omega_{nva} = \omega_{mutp} = \omega_{uva} = 1, \omega_{ncp} = 0$. For the pre-training phase, we first split the news set into 4:1 as the training set and validation set (used for early-stopping) respectively, and use them to train the news pre-trainer. We then split the user set in the same way and train the user pre-trainer. In the plug-and-play phase for CTR prediction, we follow the same splitting strategy as in Wu et al. (2020) on the MIND dataset. For the Adressa dataset, we perform 1:4 negative sampling since it has only positive interaction samples. For Adressa-1Week,

Table 3: Performance comparison among different approaches.

Method		End-to-end							Pretrain-finetune PLM		Plug-and-play PLM	
		PNN	DeepFM	DCN	NAML	LSTUR	NRMS	FIM	UNBERT	NRMS$_{PLM\text{-}NR}$	DCN$_{BERT}$	DCN$_{PREC}$ (ours)
MIND-small	AUC	61.80	61.86	65.07	**66.12**	65.87	65.63	65.34	**67.62**	/	66.12	**67.47**
	MRR	30.11	29.77	**33.12**	31.53	30.78	30.96	30.64	**31.72**		33.53	**34.88**
	nDCG@5	30.65	30.15	34.02	**34.88**	33.95	34.13	33.61	**34.75**		34.77	**36.20**
	nDCG@10	36.93	36.74	40.06	**41.09**	40.15	40.52	40.16	**41.02**		40.95	**42.33**
MIND-large	AUC	66.33	67.00	66.52	66.46	67.08	67.66	**67.87**	**70.68**	70.64	68.46	**69.12**
	MRR	32.32	32.77	32.32	32.75	32.36	33.25	**33.46**	**35.68**	35.39	34.05	**34.47**
	nDCG@5	35.11	35.59	35.08	35.66	35.15	36.28	**36.53**	**39.13**	38.71	37.17	**37.70**
	nDCG@10	40.83	41.30	40.83	41.40	40.93	41.98	**42.21**	**44.78**	44.38	42.82	**43.35**
		PNN	DeepFM	DCN	AutoInt	xDeepFM	FiBiNET	GNUD			DCN$_{BERT}$	DCN$_{PREC}$ (ours)
Adressa-1Week	AUC	71.21	73.40	75.86	71.80	**75.99**	71.21	72.03	/	/	77.30	**83.78**
	MRR	92.05	92.43	**93.26**	91.90	93.18	91.08	92.24			93.66	**95.61**
	nDCG@5	94.67	94.43	**95.30**	93.74	95.02	91.19	94.78			95.70	**97.31**
	nDCG@10	94.83	94.83	**95.51**	94.31	95.35	92.82	94.91			95.83	**97.32**
Adressa-4Week	AUC	68.81	70.21	**71.55**	69.41	71.08	66.71	69.03	/	/	74.18	**79.09**
	MRR	90.71	90.97	**91.80**	91.21	91.33	89.99	91.62			92.47	**93.90**
	nDCG@5	92.35	91.51	94.03	93.54	92.76	91.42	**95.28**			94.59	**95.67**
	nDCG@10	93.03	92.72	94.32	93.86	93.47	92.30	**95.04**			94.70	**95.86**

we use the first 5 days' interaction data as user history, the 6-th day's data as training set, and the 7-th day's data for validation (by randomly sampling 20% data) and testing (the remaining 80% data), respectively. For Adressa-4Week, we construct user history with the first 24 days' data; the following 2 days' data are used as the training set, and the 20% and 80% of the last 2 days' data are used for validation and testing, respectively. The statistics of the datasets are summarized in Table 1.

4.1.1 Evaluation Protocols

We follow common practice Wu et al. (2020, 2021a) to evaluate the effectiveness of our method with the widely-used ranking metrics: AUC, MRR, nDCG@5, and nDCG@10.

4.1.2 Implementation Details

We tokenize the title, abstract, and text descriptions in MIND and Adressa with the vocabulary provided by BERT and NordicBERT[1], respectively. Note that entities and locations are special words, so we do not tokenize them. We follow the setting of BERT (Devlin et al., 2019) and output 768-dimensional user and news vectors. For the news pre-trainer, we use 3 Transformer layers. For the user pre-trainer, we use 6 Transformer layers. We initialize the parameters as in BERT wherever possible. We apply the open-source FuxiCTR (Zhu et al., 2021) library for implementing downstream CTR prediction tasks. We release the source code for reproducibility[2].

As illustrated in Figure 2, we make a fair comparison of the running time of NRMS$_{PLM\text{-}NR}$ and DCN$_{PREC}$. The number of Transformer layers, attention heads, and dimension size of BERT (in PLM-NR), news and user pre-trainer (in PREC) are set to 12, 12, and 768, respectively. The news title and user sequence length are set to 50 and 25 respectively. We record the model updating time on a single NVIDIA GeForce RTX 3090 device.

4.1.3 Compared Models

We take 13 existing models as our baselines, including typical CTR prediction models (i.e., DCN (Wang et al., 2017), DeepFM (Guo et al., 2018), PNN (Qu et al., 2016) and xDeepFM (Lian et al., 2018), AutoInt (Song et al., 2019), FiBiNET (Huang et al., 2019)), neural news recommendation models (i.e., LSTUR (An et al., 2019), NAML (Wu et al., 2019a), NRMS (Wu et al., 2019b), GNUD (Hu et al., 2020), FIM (Wang et al., 2020)), and PLM-based models (i.e., UNBERT (Zhang et al., 2021a), PLM-NR (Wu et al., 2021a)). We also compare with the naive-BERT model with 3 Transformer layers (same as ours) to extract the news representations for DCN, denoted as DCN$_{BERT}$. In this setting, only the title view is used and the masked language modeling task is applied in news content pre-training. The repre-

[1] https://github.com/certainlyio/nordic_bert

[2] https://github.com/Jyonn/PREC

Table 4: Ablation results on the Adressa-1Week dataset. * indicates that only one mask identifier ⟨MASK⟩ is used for different views. The *exp.* column indicates experiments with different settings.

	View in NP			NP		UP		Method	
exp.	title	desc	key	MNTP	NVA	MUTP	UVA	DCN	PNN
a	-	-	-	-	-	-	-	75.86	71.21
b	✓	✓	✓	✓	-	-	-	80.06	77.82
c	✓	✓	✓	✓	✓	-	-	82.39	79.64
d	✓	✓	✓	✓	✓	✓	-	82.95	80.73
e	✓	✓	✓	✓	✓	✓	✓	83.78	81.92
f	✓	-	-	✓	-	-	-	77.30	74.77
g	✓	✓	-	✓	-	-	-	78.84	76.08
h	✓	✓	✓	*	-	-	-	79.47	77.25

sentations generated by the naive-BERT model are fixed and not fine-tuned in the downstream task. In particular, we apply PREC on DCN, which is widely used in industry, to validate its effectiveness and efficiency. We also test PREC on DeepFM and PNN to demonstrate its broad applicability.

4.2 Performance Comparison

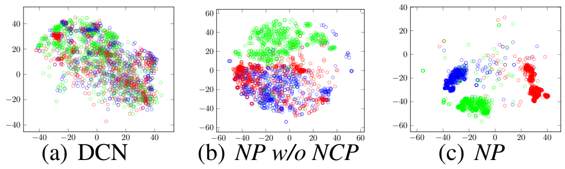
Table 2 shows the comparison between the baseline models and PREC- boosted models, where $_{+NP}$ means news embeddings are generated by the news pre-trainer and $_{+UP}$ means user embeddings are generated by the user pre-trainer. We can observe that using pre-trained news and user representations can enhance the performance of each downstream CTR model. Table 3 provides the comprehensive results of different models on four datasets. The results show that: 1) Our PREC-based model achieves competitive results with the pretrain-finetune PLM-based models, which achieve state-of-the-art performance due to the fine-tuning process and cannot be deployed in the ranking stage. 2) With the pre-trained representations as input features, the lightweight DCN model can outperform sophisticated neural news recommendation models such as FIM and GNUD, showing the effectiveness of our approach.

4.3 Ablation Study

As demonstrated in Table 4, we explore different variants during PREC pre-training. We find that 1) based on *exp. a to e*, each pre-training task makes a considerable improvement; 2) based on *exp. bfg*, each view offers better comprehension to the news pre-trainer; 3) compared with applying one mask identifier (*exp. h*), employing the view-specific mask identifier (*exp. b*) achieves better performance. We also conduct experiments on the effect of the number of Transformer layers. As depicted in Table 5, the performance improves when the number of layers increases, since more Transformer layers can capture deeper semantic meanings. However, the pre-training time cost also grows. Hence, we choose 3 layers as a trade-off between performance and efficiency.

Table 5: Influence of the Transformer layers of the news pre-trainer on the Adressa-1Week dataset. Time (h) denotes the pre-training time in hours.

Layers of NP	1	3	6	9	12
AUC	81.34	82.39	82.60	83.02	83.66
Time (h)	12.0	13.3	14.0	15.9	18.3
Improv. per Layer	-	**0.525**	0.252	0.210	0.211



(a) DCN (b) *NP w/o NCP* (c) *NP*

Figure 5: Visualization of the news embeddings of the MIND-small dataset. Different color indicates different news category.

4.4 Visualization of News Embeddings

We use t-SNE (Van der Maaten and Hinton, 2008) to visualize the news embeddings learned by different settings. As depicted in Figure 5, we randomly select three categories and observe the embedding distribution. It can be seen that the news embeddings learned by the DCN model (Figure 5(a)) are scattered while the news features obtained by the news pre-trainer (Figure 5(b)) are clustered. When the NCP task is adopted Figure 5(c), the news features are more clustered.

4.5 Online A/B Testing

We have deployed our news pre-trainer in Huawei's news recommender system, serving millions of users daily, to perform an online A/B test. CTR prediction is applied in the ranking phase of recommendation, which takes tens of user features (e.g., city, tags clicked in recent 1/3/7 days) and news features (e.g., category, topic, tags, entities) as input and outputs the predicted click probability of each user-news pair. The base model deployed online is an optimized variant of the DeepFM model (Guo et al., 2018) called FINAL and has a stringent latency requirement (less than 50ms) for each request. To improve the model performance, yet keep the efficiency of model inference, we apply the PREC pre-trainer to obtain cached representation vectors of users and news. Then, these vectors are used to initialize the embedding layers for training the ranking model. The decoupling allows asynchronous updates of both modules: the PREC pre-trainer is updated on a daily basis while the ranking model is updated on a minute level to adapt to new data quickly. During the one-week online A/B test, the PREC-boosted ranking model has achieved an average improvement of 2.4% in CTR over the baseline, which is significant in our application scenario.

5 Conclusion

In this paper, we have developed a plug-and-play pre-trainer called PREC to boost the performance of traditional ID-based CTR models for news recommendation. PREC is built on Transformer layers, utilizes multi-view features of news and users, and is trained with tailored pre-training tasks to learn semantic news and user representations. Aside from its ability in content understanding, PREC can be easily deployed in industrial recommender systems to improve CTR prediction. Offline experiments on public benchmark datasets and online A/B testing in industrial recommender systems demonstrate the effectiveness and efficiency of PREC.

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful comments.

References

Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural news recommendation with long- and short-term user representations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 336–345, Florence, Italy. Association for Computational Linguistics.

Xusong Chen, Dong Liu, Chenyi Lei, Rui Li, Zheng-Jun Zha, and Zhiwei Xiong. 2019. Bert4sessrec: Content-based video relevance prediction with bidirectional encoder representations from transformer. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2597–2601.

Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Jon Atle Gulla, Lemei Zhang, Peng Liu, Özlem Özgöbek, and Xiaomeng Su. 2017. The adressa dataset for news recommendation. In *Proceedings of the international conference on web intelligence*, pages 1042–1048.

Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, et al. 2018. Deepfm: An end-to-end wide & deep learning framework for ctr prediction. In *International Joint Conferences on Artificial Intelligence*.

Linmei Hu, Siyong Xu, Chen Li, Cheng Yang, Chuan Shi, Nan Duan, Xing Xie, and Ming Zhou. 2020. Graph neural news recommendation with unsupervised preference disentanglement. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 4255–4264.

Tongwen Huang, Zhiqi Zhang, and Junlin Zhang. 2019. Fibinet: combining feature importance and bilinear feature interaction for click-through rate prediction. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 169–177.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Jian Li, Jieming Zhu, Qiwei Bi, Guohao Cai, Lifeng Shang, Zhenhua Dong, Xin Jiang, and Qun Liu. 2022. Miner: Multi-interest matching network for news recommendation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 343–352.

Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1754–1763.

Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *The International Conference on Learning Representations*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-based neural networks for user response prediction. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 1149–1154. IEEE.

Suvash Sedhain, Scott Sanner, Darius Braziunas, Lexing Xie, and Jordan Christensen. 2014. Social collaborative filtering for cold-start recommendations. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 345–348.

Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1161–1170.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Heyuan Wang, Fangzhao Wu, Zheng Liu, and Xing Xie. 2020. Fine-grained interest matching for neural news recommendation. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 836–845.

Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. Dkn: Deep knowledge-aware network for news recommendation. In *Proceedings of the 10th international conference on World Wide Web*.

Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep amp; cross network for ad click predictions. In *Proceedings of the ADKDD'17*, ADKDD'17, New York, NY, USA. Association for Computing Machinery.

Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, et al. 2019a. Neural news recommendation with attentive multi-view learning. In *International Joint Conferences on Artificial Intelligence*.

Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019b. Neural news recommendation with multi-head self-attention. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 6389–6394.

Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021a. Empowering news recommendation with pre-trained language models. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1652–1656.

Chuhan Wu, Fangzhao Wu, Yang Yu, Tao Qi, Yongfeng Huang, and Qi Liu. 2021b. NewsBERT: Distilling pre-trained language model for intelligent news application. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3285–3295, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chuhan Wu, Fangzhao Wu, Yang Yu, Tao Qi, Yongfeng Huang, and Xing Xie. 2021c. Userbert: Contrastive user model pre-training. *arXiv preprint arXiv:2109.01274*.

Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, et al. 2020. Mind: A large-scale dataset for news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3597–3606.

Shitao Xiao, Zheng Liu, Yingxia Shao, Tao Di, Bhuvan Middha, Fangzhao Wu, and Xing Xie. 2022. Training large-scale news recommenders with pretrained language models in the loop. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4215–4225.

Xu Xie, Fei Sun, Zhaoyang Liu, Jinyang Gao, Bolin Ding, and Bin Cui. 2020. Contrastive pre-training for sequential recommendation. *arXiv preprint arXiv:2010.14395*.

Yang Yu, Fangzhao Wu, Chuhan Wu, Jingwei Yi, Tao Qi, and Qi Liu. 2021. Tiny-newsrec: Efficient and effective plm-based news recommendation. In *ArXiv*.

Qi Zhang, Jingjie Li, Qinglin Jia, Chuyuan Wang, et al. 2021a. Unbert: User-news matching bert for news recommendation. In *International Joint Conferences on Artificial Intelligence*.

Weinan Zhang, Jiarui Qin, Wei Guo, Ruiming Tang, and Xiuqiang He. 2021b. Deep learning for click-through rate estimation. In *International Joint Conferences on Artificial Intelligence*.

Kun Zhou, 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*, pages 1893–1902.

Jieming Zhu, Jinyang Liu, Shuai Yang, Qi Zhang, and Xiuqiang He. 2021. Open benchmarking for click-through rate prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2759–2769.