Tiny-NewsRec: Effective and Efficient PLM-based News Recommendation

Yang Yu[1], **Fangzhao Wu**[2], **Chuhan Wu**[3], **Jingwei Yi**[1] **and Qi Liu**[1]
[1]University of Science and Technology of China
[2]Microsoft Research Asia [3]Tsinghua University
{yflyl613, yjw1029}@mail.ustc.edu.cn
{wufangzhao, wuchuhan15}@gmail.com, qiliuql@ustc.edu.cn

Abstract

News recommendation is a widely adopted technique to provide personalized news feeds for the user. Recently, pre-trained language models (PLMs) have demonstrated the great capability of natural language understanding and benefited news recommendation via improving news modeling. However, most existing works simply finetune the PLM with the news recommendation task, which may suffer from the known domain shift problem between the pre-training corpus and downstream news texts. Moreover, PLMs usually contain a large volume of parameters and have high computational overhead, which imposes a great burden on low-latency online services. In this paper, we propose Tiny-NewsRec, which can improve both the effectiveness and the efficiency of PLM-based news recommendation. We first design a self-supervised domain-specific post-training method to better adapt the general PLM to the news domain with a contrastive matching task between news titles and news bodies. We further propose a two-stage knowledge distillation method to improve the efficiency of the large PLM-based news recommendation model while maintaining its performance. Multiple teacher models originated from different time steps of our post-training procedure are used to transfer comprehensive knowledge to the student in both its post-training and finetuning stage. Extensive experiments on two real-world datasets validate the effectiveness and efficiency of our method.

1 Introduction

With the explosion of information, massive news is published on online news platforms such as Microsoft News and Google News (Das et al., 2007; Lavie et al., 2010), which can easily get the users overwhelmed when they try to find the information they are interested in (Okura et al., 2017; Li et al., 2020b). Many personalized news recommendation methods have been proposed to alleviate the information overload problem for users (Wang et al., 2018; Wu et al., 2019b; Zhu et al., 2019; Hu et al., 2020). Since news articles usually contain abundant textual content, learning high-quality news representations from news texts is one of the most critical tasks for news recommendation (Wu et al., 2020). As pre-trained language models (PLMs) have been proved to be powerful in text modeling and have empowered various NLP tasks (Devlin et al., 2019; Liu et al., 2019), a few recent works delve into employing PLMs for better news modeling in news recommendation (Wu et al., 2021b; Jia et al., 2021; Zhang et al., 2021b). For example, Wu et al. (2021b) propose to replace shallow NLP models such as CNN and attention network with the PLM to capture the deep contexts in news texts. However, these methods simply finetune the PLM with the news recommendation task, which may be insufficient to cope with the domain shift problem between the generic pre-training corpus and downstream news texts (Gururangan et al., 2020; Madan et al., 2021). Moreover, PLMs usually have a large number of parameters. For example, the BERT-base model (Devlin et al., 2019) contains 12 layers with 110M parameters. Deploying these PLM-based news recommendation models to provide low-latency online services requires extensive computational resources.

In this paper, we propose a Tiny-NewsRec approach to improve both the effectiveness and the efficiency of PLM-based news recommendation[1]. In our approach, we first utilize the natural matching relation between different parts of a news article and design a self-supervised domain-specific post-training method to better adapt the general PLM to the news domain. The PLM-based news encoder is trained with a contrastive matching task between news titles and news bodies to make it better capture the semantic information in news texts and generate more discriminative representations, which

[1] The source code and data of our Tiny-NewsRec are available at https://github.com/yflyl613/Tiny-NewsRec.

are beneficial to both news content understanding and user interest matching in the following news recommendation task. In addition, we propose a two-stage knowledge distillation method to compress the large PLM-based model while maintaining its performance[2]. Domain-specific knowledge and task-specific knowledge are transferred from the teacher model to the student in its post-training stage and finetuning stage respectively. Besides, multiple teacher models originated from different time steps of our post-training procedure are used to provide comprehensive guidance to the student model in both stages. For each training sample, we adaptively weight these teacher models based on their performance, which allows the student model to always learn more from the best teacher. Extensive experiment results on two real-world datasets show that our approach can reduce the model size by 50%-70% and accelerate the inference speed by 2-8 times while achieving better performance. The main contributions of our paper are as follows:

- We propose a Tiny-NewsRec approach to improve both the effectiveness and efficiency of PLM-based news recommendation.
- We propose a self-supervised domain-specific post-training method which trains the PLM with a contrastive matching task between news titles and news bodies before the task-specific finetuning to better adapt it to the news domain.
- We propose a two-stage knowledge distillation method with multiple teacher models to compress the large PLM-based model.
- Extensive experiments on two real-world datasets validate that our method can effectively improve the performance of PLM-based news recommendation models while reducing the model size by a large margin.

2 Related Work

2.1 PLM-based News Recommendation

With the great success of pre-trained language models (PLMs) in multiple NLP tasks, many researchers have proposed to incorporate the PLM in news recommendation and have achieved substantial gain (Zhang et al., 2021b; Jia et al., 2021; Wu et al., 2021b). For example, Zhang et al. (2021b) proposed UNBERT, which utilizes the PLM to capture multi-grained user-news matching signals at both word-level and news-level. Wu et al. (2021b) proposed a state-of-the-art PLM-based news recommendation method named PLM-NR, which instantiates the news encoder with a PLM to capture the deep semantic information in news texts and generate high-quality news representations. However, these methods simply finetune the PLM with the news recommendation task, the supervision from which may be insufficient to fill the domain gap between the generic pre-training corpus and downstream news texts (Gururangan et al., 2020; Madan et al., 2021). Besides, PLMs usually contain a large number of parameters and have high computational overhead. Different from these methods, our approach can better mitigate the domain shift problem with an additional domain-specific post-training task and further reduce the computational cost with a two-stage knowledge distillation method.

2.2 Domain Adaptation of the PLM

Finetuning a PLM has become a standard procedure for many NLP tasks (Devlin et al., 2019; Raffel et al., 2020). These models are first pre-trained on large generic corpora (e.g., BookCorpus and Wikipedia) and then finetuned on the downstream task data. Even though this paradigm has achieved great success, it suffers from the known domain shift problem between the pre-training and downstream corpus (Howard and Ruder, 2018; Lee et al., 2019; Beltagy et al., 2019). A technique commonly used to mitigate this problem is continuing to pre-train the general PLM on additional corpora related to the downstream task (Logeswaran et al., 2019; Chakrabarty et al., 2019; Han and Eisenstein, 2019). For example, Gururangan et al. (2020) proposed domain-adaptive pre-training (DAPT) and task-adaptive pre-training (TAPT), which further pre-trains the PLM on a large corpus of unlabeled domain-specific text and the training text set for a given task before the task-specific finetuning, respectively. Instead of continued pre-training, we utilize the natural matching relation between different parts of a news article and design a domain-specific post-training method with a contrastive matching task between news titles and news bodies. It can make the PLM better capture the high-level semantic information in news texts and generate more discriminative news representations, which are beneficial for news recommendation.

2.3 PLM Knowledge Distillation

Knowledge distillation (KD) is a technique that aims to compress a heavy teacher model into a

[2] We focus on task-specific knowledge distillation.

lightweight student model while maintaining its performance (Hinton et al., 2015). In recent years, many works explore compressing large-scale PLMs via KD (Sun et al., 2019; Wang et al., 2020; Sun et al., 2020; Xu et al., 2020). For example, Jiao et al. (2020) proposed TinyBERT, which lets the student model imitate the intermediate and final outputs of the teacher model in both the pre-training and fine-tuning stages. There are also a few works that aim to distill the PLM for specific downstream tasks (Lu et al., 2020; Wu et al., 2021c). For example, Wu et al. (2021c) proposed NewsBERT for intelligent news applications. A teacher-student joint distillation framework is proposed to collaboratively learn both teacher and student models. Considering that the guidance provided by a single teacher may be limited or even biased, some works propose to conduct KD with multiple teacher models (Liu et al., 2020; Wu et al., 2021a). However, all these works neglect the potential domain gap between the pre-training corpus and the downstream task domain. To our best knowledge, we are the first to conduct KD during the domain adaptation of PLMs. Both domain-specific and task-specific knowledge are transferred to the student model in our two-stage knowledge distillation method. Besides, multiple teacher models are used to provide more comprehensive guidance to the student in both stages.

3 Methodology

In this section, we introduce the details of our Tiny-NewsRec method. We first briefly introduce the structure of our PLM-based news recommendation model. Then we introduce the design of our self-supervised domain-specific post-training method and the framework of our two-stage knowledge distillation method. Some notations used in the paper are listed in Table 1.

3.1 News Recommendation Model

We first introduce the structure of the PLM-based news recommendation model used in our Tiny-NewsRec. As shown in Fig. 1(b), it consists of three major components, i.e., a news encoder, a user encoder, and a click prediction module. The news encoder aims to learn the news representation from news texts. Following the state-of-the-art PLM-based news recommendation method (Wu et al., 2021b), we use a PLM to capture the deep context in news texts and an attention network to aggregate the output of the PLM. The user encoder aims to learn the user representation from the representations of the user's last L clicked news, i.e., $[\boldsymbol{n}_1, \boldsymbol{n}_2, ..., \boldsymbol{n}_L]$. Following Wu et al. (2019a), we implement it with an attention network to select important news from the user's historical interactions. In the click prediction module, we take the dot product of the candidate news representation $\boldsymbol{n}_c$ and the target user representation $\boldsymbol{u}$ as the predicted score $\hat{y}_{\mathrm{FT}}$. It is noted that our Tiny-NewsRec is decoupled from the structure of the news recommendation model. Other PLM-based news recommendation models (Jia et al., 2021; Zhang et al., 2021a,b) can also be adopted.

Notation	Explanation
$\boldsymbol{h}_{nb}$	News body representation
$\boldsymbol{h}_{nt}$	News title representation
$\boldsymbol{n}$	News representation
$\boldsymbol{u}$	User representation
$\mathrm{CE}(\cdot,\cdot)$	Cross-Entropy loss function
$\mathrm{MSE}(\cdot,\cdot)$	Mean-Squared Error loss function
$^{(t_i)}$	Outputs or parameters of the i-th teacher model
$^{(s)}$	Outputs or parameters of the student model
FT	Abbreviation for "Finetune"
DP	Abbreviation for "Domain-specific Post-train"

Table 1: Some notations used in this paper.

3.2 Domain-specific Post-training

Since directly finetuning the PLM with the downstream news recommendation task may be insufficient to fill the domain gap between the general corpus and news texts (Gururangan et al., 2020; Madan et al., 2021), we propose to conduct domain-specific post-training to the PLM before the task-specific finetuning. Considering the natural matching relation between different parts of a news article, we design a self-supervised contrastive matching task between news titles and news bodies. The model framework for this task is shown in Fig. 1(a).

Given a news article, we regard its news body nb as the anchor and take its news title nt^+ as the positive sample. We randomly select N other news titles $[nt_1^-, nt_2^-, \cdots, nt_N^-]$ from the news pool as negative samples. We use the PLM-based news encoder to get the news body representation $\boldsymbol{h}_{nb}$ and these news title representations $[\boldsymbol{h}_{nt^+}, \boldsymbol{h}_{nt_1^-}, \boldsymbol{h}_{nt_2^-}, \cdots, \boldsymbol{h}_{nt_N^-}]$. We adopt the InfoNCE loss (Oord et al., 2018) as the contrastive loss function. It is formulated as follows:

$$\mathcal{L}_{\mathrm{DP}} = -\log \frac{\exp(\hat{y}_{nt^+})}{\exp(\hat{y}_{nt^+}) + \sum_{i=1}^{N} \exp(\hat{y}_{nt_i^-})},$$

where $\hat{y}_{nt^+} = \boldsymbol{h}_{nb}^{\mathrm{T}} \boldsymbol{h}_{nt^+}$ and $\hat{y}_{nt_i^-} = \boldsymbol{h}_{nb}^{\mathrm{T}} \boldsymbol{h}_{nt_i^-}$. As

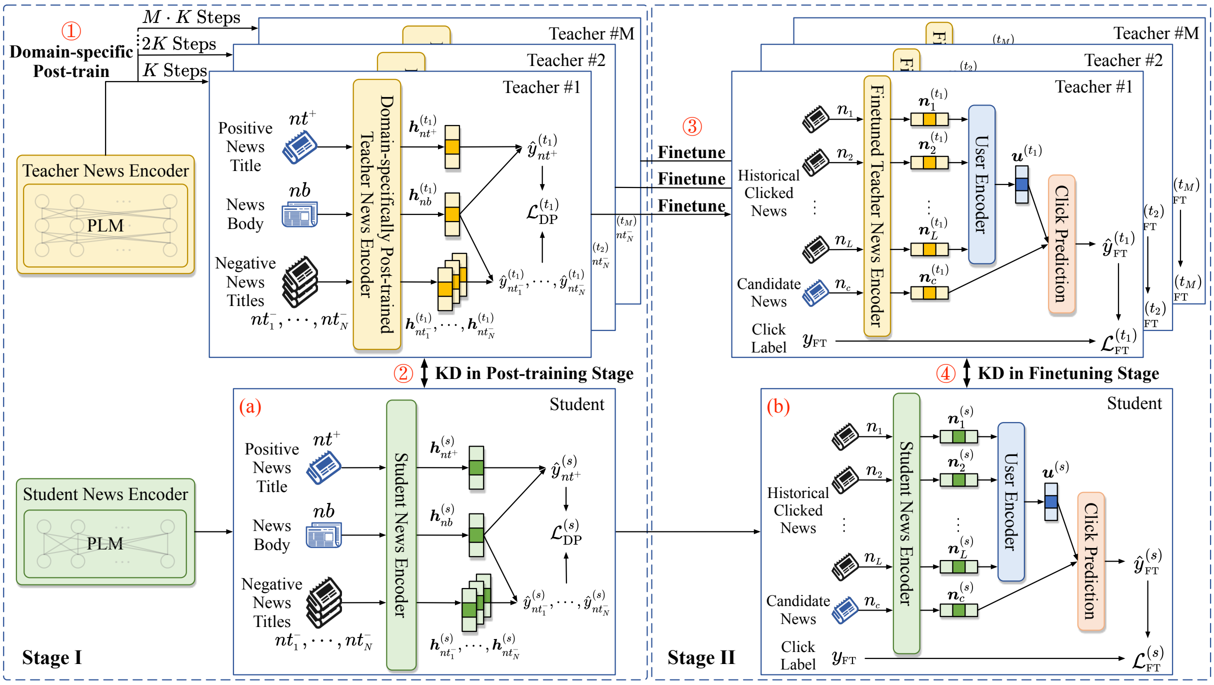

Figure 1: The framework of Tiny-NewsRec

proved by Oord et al. (2018), minimizing $\mathcal{L}_{\mathrm{DP}}$ can maximize the lower bound of the mutual information between $\boldsymbol{h}_{nb}$ and $\boldsymbol{h}_{nt^+}$. Therefore, the post-trained PLM-based news encoder can better capture and match the high-level semantic information in news texts. It will generate more similar representations for related texts (i.e., the news body and its corresponding news title) and distinguish them from the others, which can also ease the anisotropy problem of the sentence representation generated by the PLM (Gao et al., 2019; Ethayarajh, 2019; Li et al., 2020a). Thus, our proposed domain-specific post-training method is beneficial to both news understanding and user interest matching in the following news recommendation task.

3.3 Two-stage Knowledge Distillation

To achieve our goal of efficiency, we further propose a two-stage knowledge distillation method, whose framework is shown in Fig. 1. In our framework, the lightweight student model is trained to imitate the large teacher model in both its post-training stage and finetuning stage. Besides, multiple teacher models originated from different time steps of our post-training procedure are used to transfer more comprehensive knowledge to the student model in both stages.

In Stage I, we first conduct domain-specific post-training towards the teacher PLM-based news encoder (Step 1). During the post-training procedure, a copy of the current teacher news encoder is saved every K steps after convergency and we save M teacher models in total. Then we use these teacher models to transfer comprehensive domain-specific knowledge to the student model during its post-training (Step 2). Since these teacher models at different time steps may have different performance on an input sample, we assign an adaptive weight to each teacher for each training sample, which is measured by the cross-entropy loss between its predicted scores $\hat{\boldsymbol{y}}_{\mathrm{DP}}^{(t_i)} = [\hat{y}_{nt^+}^{(t_i)}, \hat{y}_{nt_1^-}^{(t_i)}, \hat{y}_{nt_2^-}^{(t_i)}, \cdots, \hat{y}_{nt_N^-}^{(t_i)}]$ and the ground-truth label y_{DP}. Denote the weight of the i-th teacher model on a given sample as $\alpha^{(t_i)}$, it is formulated as follows:

$$\alpha^{(t_i)} = \frac{\exp(-\mathrm{CE}(\hat{\boldsymbol{y}}_{\mathrm{DP}}^{(t_i)}, y_{\mathrm{DP}}))}{\sum_{j=1}^{M}\exp(-\mathrm{CE}(\hat{\boldsymbol{y}}_{\mathrm{DP}}^{(t_j)}, y_{\mathrm{DP}}))}.$$

To encourage the student model to make similar predictions to the best teacher model, we use a distillation loss to regularize its output soft labels, which is formulated as follows:

$$\mathcal{L}_{\mathrm{DP}}^{\mathrm{distill}} = T_{\mathrm{DP}}^2 \cdot \mathrm{CE}(\sum_{i=1}^{M} \alpha^{(t_i)} \hat{\boldsymbol{y}}_{\mathrm{DP}}^{(t_i)}/T_{\mathrm{DP}}, \hat{\boldsymbol{y}}_{\mathrm{DP}}^{(s)}/T_{\mathrm{DP}}).$$

T_{DP} is a temperature hyper-parameter that controls the smoothness of the predicted probability distribution of the teacher models. Besides, since we expect the representations generated by the student model and these teacher models to be similar in a unified space, we propose to apply an additional embedding loss to align these representations. The embedding loss between the i-th teacher model and the student model is formulated as follows:

$$\begin{aligned}\mathcal{L}_{\text{DP}}^{\text{emb}_i} = &\text{MSE}(\boldsymbol{W}^{(t_i)}\boldsymbol{h}_{nt}^{(t_i)} + \boldsymbol{b}^{(t_i)}, \boldsymbol{h}_{nt}^{(s)}) + \\ &\text{MSE}(\boldsymbol{W}^{(t_i)}\boldsymbol{h}_{nb}^{(t_i)} + \boldsymbol{b}^{(t_i)}, \boldsymbol{h}_{nb}^{(s)}),\end{aligned}$$

where $\boldsymbol{W}^{(t_i)}$ and $\boldsymbol{b}^{(t_i)}$ are the learnable parameters in the additional linear projection layer of the i-th teacher model. The overall embedding loss is the weighted summation of all these embedding losses, i.e., $\mathcal{L}_{\text{DP}}^{\text{emb}} = \sum_{i=1}^{M} \alpha^{(t_i)} \mathcal{L}_{\text{DP}}^{\text{emb}_i}$. The loss function for the student model in Stage I is the summation of the distillation loss, the overall embedding loss, and its InfoNCE loss in our domain-specific post-training task, which is formulated as follows:

$$\mathcal{L}_1 = \mathcal{L}_{\text{DP}}^{\text{distill}} + \mathcal{L}_{\text{DP}}^{\text{emb}} + \mathcal{L}_{\text{DP}}^{(s)}.$$

Next, in Stage II, we first finetune these M post-trained teacher news encoders with the news recommendation task (Step 3). Then they are used to transfer rich task-specific knowledge to the student during its finetuning (Step 4). Similar to Stage I, we assign a weight $\beta^{(t_i)}$ to each finetuned teacher model based on its cross-entropy loss given an input sample of the news recommendation task and apply the following distillation loss to adjust the output of the student model during its finetuning:

$$\beta^{(t_i)} = \frac{\exp(-\,\text{CE}(\hat{\boldsymbol{y}}_{\text{FT}}^{(t_i)}, y_{\text{FT}}))}{\sum_{j=1}^{M} \exp(-\,\text{CE}(\hat{\boldsymbol{y}}_{\text{FT}}^{(t_j)}, y_{\text{FT}}))},$$

$$\mathcal{L}_{\text{FT}}^{\text{distill}} = T_{\text{FT}}^2 \cdot \text{CE}(\sum_{i=1}^{M} \beta^{(t_i)} \hat{\boldsymbol{y}}_{\text{FT}}^{(t_i)} / T_{\text{FT}}, \hat{\boldsymbol{y}}_{\text{FT}}^{(s)} / T_{\text{FT}}),$$

where $\hat{\boldsymbol{y}}_{\text{FT}}$ denotes the predicted score of the model on the news recommendation task and T_{FT} is another temperature hyper-parameter. We also use an additional embedding loss to align both the news representation and the user representation of the student model and the teacher models, which is formulated as follows:

$$\begin{aligned}\mathcal{L}_{\text{FT}}^{\text{emb}} = \sum_{i=1}^{M} \beta^{(t_i)} [&\text{MSE}(\boldsymbol{W}_n^{(t_i)}\boldsymbol{n}^{(t_i)} + \boldsymbol{b}_n^{(t_i)}, \boldsymbol{n}^{(s)}) + \\ &\text{MSE}(\boldsymbol{W}_u^{(t_i)}\boldsymbol{u}^{(t_i)} + \boldsymbol{b}_u^{(t_i)}, \boldsymbol{u}^{(s)})],\end{aligned}$$

where $\boldsymbol{W}_n^{(t_i)}$, $\boldsymbol{b}_n^{(t_i)}$ and $\boldsymbol{W}_u^{(t_i)}$, $\boldsymbol{b}_u^{(t_i)}$ are the learnable parameters used to project the news representations and the user presentations learned by the i-th teacher model into a unified space, respectively.

	MIND		
# News	161,013	# Users	1,000,000
# Impressions	15,777,377	# Clicks	24,155,470
Avg. title length	11.52		
	Feeds		
# News	377,296	# Users	10,000
# Impressions	320,925	# Clicks	437,072
Avg. title length	11.93		
	News		
# News	1,975,767	Avg. title length	11.84
Avg. body length	511.43		

Table 2: Detailed statistics of *MIND*, *Feeds* and *News*.

The student model is also tuned to minimize the cross-entropy loss between its predicted score $\hat{\boldsymbol{y}}_{\text{FT}}^{(s)}$ and the ground-truth label y_{FT} of the news recommendation task, i.e., $\mathcal{L}_{\text{FT}}^{(s)} = \text{CE}(\hat{\boldsymbol{y}}_{\text{FT}}^{(s)}, y_{\text{FT}})$. The overall loss function for the student model in Stage II is the summation of the distillation loss, the embedding loss, and its finetuning loss, which is formulated as follows:

$$\mathcal{L}_2 = \mathcal{L}_{\text{FT}}^{\text{distill}} + \mathcal{L}_{\text{FT}}^{\text{emb}} + \mathcal{L}_{\text{FT}}^{(s)}.$$

4 Experiments

4.1 Datasets and Experimental Settings

We conduct experiments with three real-world datasets, i.e., *MIND*, *Feeds*, and *News*. *MIND* is a public dataset for news recommendation (Wu et al., 2020), which contains the news click logs of 1,000,000 users on the Microsoft News website in six weeks. We use its public training set, validation set, and test set for experiments[3]. *Feeds* is also a news recommendation dataset collected on the Microsoft News App from 2020-08-01 to 2020-09-01. We use the impressions in the last week for testing and randomly sampled 20% impressions from the training set for validation. *News* contains news articles collected on the Microsoft News website from 2020-09-01 to 2020-10-01, which is used for our domain-specific post-training task. Detailed statistics of these datasets are summarized in Table 2.

In our experiments, following PLM-NR (Wu et al., 2021b), we apply the pre-trained UniLMv2 (Bao et al., 2020) to initialize the PLM in the news encoder due to its superior text modeling capability. The dimensions of the news representation and the user representation are both 256. The temperature hyper-parameters T_{DP} and T_{FT} are both set to 1. A copy of the teacher model is saved every $K = 500$ steps during post-training and the number of teacher models M is set to 4. We use the

[3] We randomly choose 1/2 samples from the original training set as our training data due to the limit of training speed.

Model	MIND			Feeds			Model
	AUC	MRR	nDCG@10	AUC	MRR	nDCG@10	Size
PLM-NR$_{12}$ (FT)	69.72±0.15	34.74±0.10	43.71±0.07	67.93±0.13	34.42±0.07	45.09±0.07	109.89M
PLM-NR$_{12}$ (DAPT)	69.97±0.08	35.07±0.15	43.98±0.10	68.24±0.09	34.63±0.10	45.30±0.09	109.89M
PLM-NR$_{12}$ (TAPT)	69.82±0.14	34.90±0.11	43.83±0.07	68.11±0.11	34.49±0.12	45.11±0.08	109.89M
PLM-NR$_{12}$ (DP)	**71.02±0.07**	**36.05±0.09**	**45.03±0.12**	**69.37±0.10**	**35.74±0.11**	**46.45±0.11**	109.89M
PLM-NR$_4$ (FT)	69.49±0.14	34.40±0.10	43.40±0.09	67.46±0.12	33.71±0.11	44.36±0.09	53.18M
PLM-NR$_2$ (FT)	68.99±0.08	33.59±0.14	42.61±0.11	67.05±0.14	33.33±0.09	43.90±0.12	39.01M
PLM-NR$_1$ (FT)	68.12±0.12	33.20±0.07	42.07±0.10	66.26±0.10	32.55±0.12	42.99±0.09	31.92M
TinyBERT$_4$	70.55±0.10	35.60±0.12	44.47±0.08	68.40±0.08	34.64±0.10	45.21±0.11	53.18M
TinyBERT$_2$	70.24±0.13	34.93±0.07	43.98±0.10	68.01±0.07	34.37±0.09	44.90±0.10	39.01M
TinyBERT$_1$	69.19±0.09	34.35±0.10	43.12±0.07	67.16±0.11	33.42±0.07	43.95±0.07	31.92M
NewsBERT$_4$	70.62±0.15	35.72±0.11	44.65±0.08	68.69±0.10	34.90±0.08	45.64±0.11	53.18M
NewsBERT$_2$	70.41±0.09	35.46±0.07	44.35±0.10	68.24±0.09	34.64±0.11	45.23±0.10	39.01M
NewsBERT$_1$	69.45±0.11	34.75±0.09	43.54±0.12	67.37±0.05	33.55±0.10	44.12±0.08	31.92M
Tiny-NewsRec$_4$	**71.19±0.08**	**36.21±0.05**	**45.20±0.09**	**69.58±0.06**	**35.90±0.11**	**46.57±0.07**	53.18M
Tiny-NewsRec$_2$	70.95±0.04	36.05±0.08	44.93±0.10	69.25±0.07	35.45±0.09	46.25±0.10	39.01M
Tiny-NewsRec$_1$	70.04±0.06	35.16±0.10	44.10±0.08	68.31±0.03	34.65±0.08	45.32±0.08	31.92M

Table 3: Performance comparisons of different models. The results of the best-performed teacher model and student model are highlighted. The subscript number denotes the number of layers in the model. The model size is measured by the number of parameters.

Adam optimizer (Kingma and Ba, 2015) for training. The detailed experimental settings are listed in the Appendix. All the hyper-parameters are tuned on the validation set. Following Wu et al. (2020), we use AUC, MRR, and nDCG@10 to measure the performance of news recommendation models. We independently repeat each experiment 5 times and report the average results with standard deviations.

4.2 Performance Comparison

In this section, we compare the performance of the 12-layer teacher model PLM-NR$_{12}$ (DP) which is domain-specifically post-trained before finetuning, and the student models trained with our Tiny-NewsRec with the following baseline methods:

- **PLM-NR (FT)** (Wu et al., 2021b), the state-of-the-art PLM-based news recommendation method which applies the PLM to the news encoder and directly fine-tunes it with the news recommendation task. We compare the performance of its 12-layer version and its variant using the first 1, 2, or 4 layers of the PLM.
- **PLM-NR (DAPT)**, a variant of PLM-NR which first adapts the PLM to the news domain via domain-adaptive pre-training (Gururangan et al., 2020). It continues to pre-train the PLM on a corpus of unlabeled news domain texts and then finetunes it with the news recommendation task.
- **PLM-NR (TAPT)**, another variant of PLM-NR which first adapts the PLM to the downstream task with task-adaptive pre-training (Gururangan et al., 2020). It continues to pre-train the PLM on the unlabeled news set provided along with the downstream training data and then finetunes it with the news recommendation task.
- **TinyBERT** (Jiao et al., 2020), a state-of-the-art two-stage knowledge distillation method for compressing the PLM which conducts knowledge distillation in both the pre-training stage and the finetuning stage. For a fair comparison, we use the PLM-NR$_{12}$ (DP) as the teacher model.
- **NewsBERT** (Wu et al., 2021c), a PLM knowledge distillation method specialized for intelligent news applications which jointly trains the student model and the teacher model during finetuning. For a fair comparison, we use the 12-layer domain-specifically post-trained news encoder to initialize the teacher model.

Table 3 shows the performance of all these methods on the *MIND* and *Feeds* datasets. From the results, we have the following observations. First, both PLM-NR$_{12}$ (DAPT) and PLM-NR$_{12}$ (TAPT) outperform PLM-NR$_{12}$ (FT). It validates that continued pre-training on the corpus related to the downstream task can mitigate the domain shift problem to some extent. Second, our PLM-NR$_{12}$ (DP) achieves the best performance among all 12-layer models. This is because our proposed self-supervised domain-specific post-training task can help the PLM better capture the semantic information in news texts and generate more discriminative news representations, which is beneficial to the

Model	AUC	MRR	nDCG@10
Ensemble-Teacher$_{12}$	69.43	35.81	46.53
TinyBERT-MT$_4$	68.87	35.13	45.81
NewsBERT-MT$_4$	68.82	35.07	45.80
MT-BERT$_4$	68.51	34.74	45.45
Tiny-NewsRec$_4$	**69.58**	**35.90**	**46.57**

Table 4: Performance comparisons of the ensemble teacher models and the student models distilled with various multi-teacher knowledge distillation methods.

news understanding and user interest matching in the news recommendation task. Third, compared with state-of-the-art knowledge distillation methods (i.e., NewsBERT and TinyBERT), our Tiny-NewsRec achieves the best performance in all 1-layer, 2-layer, and 4-layer student models, and our further t-test results show the improvements are significant at $p < 0.01$ (by comparing the models with the same number of layers). This is because the student model can better adapt to the news domain with supervision from the domain-specifically post-trained teacher models in Stage I, and task-specific knowledge is also transferred to it during the knowledge distillation in Stage II. Finally, our Tiny-NewsRec even achieves comparable performance with the teacher model PLM-NR$_{12}$ (DP) while having much fewer parameters and lower computational overhead. This is because these multiple teacher models originated from different time steps of the post-training procedure may complement each other and provide more comprehensive knowledge to the student model in both stages.

4.3 Further Comparison

To better understand where the performance improvement of our approach comes from, we further compare our Tiny-NewsRec with the following methods which use multiple teacher models:

- **Ensemble-Teacher**, which is the ensemble of the multiple 12-layer teacher models used by Tiny-NewsRec. The average predicted score of these teacher models is used for evaluation.
- **TinyBERT-MT** and **NewsBERT-MT**, the modified version of TinyBERT (Jiao et al., 2020) and NewsBERT (Wu et al., 2021c), which utilize the multiple teacher models used by Tiny-NewsRec. Each teacher model is adaptively weighted according to its performance on the input training sample, which is the same as the one used in our two-stage knowledge distillation method.
- **MT-BERT** (Wu et al., 2021a), which jointly finetunes the student model and multiple teacher models with different PLMs on the downstream news recommendation task.

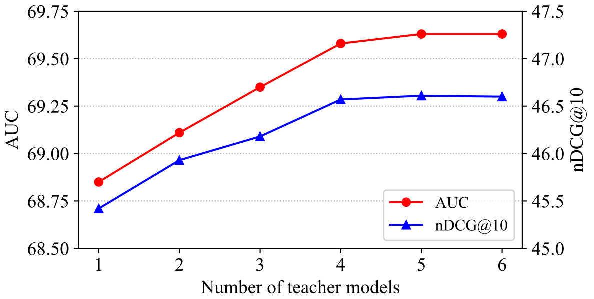

Figure 2: Impact of the number of teacher models M.

Table 4 shows the performance of the ensemble teacher models and the 4-layer student models on the *Feeds* dataset. Comparing with the results of PLM-NR$_{12}$ (DP) in Table 3, we first find that the simple ensemble of multiple teacher models cannot bring much performance gain. The reason is that these teachers are treated equally during testing. However, in our Tiny-NewsRec, for each training sample, we assign an adaptive weight to each teacher model based on their performance. The student can always learn more from the best teacher model on each sample and receive more comprehensive knowledge. Second, even with the same teacher models, Tiny-NewsRec still outperforms TinyBERT-MT and NewsBERT-MT. This is because we are the first to use multiple teacher models to transfer domain-specific knowledge to the student before the task-specific finetuning, which can help the student model better adapt to the news domain. Besides, we find that MT-BERT achieves the worst performance among all the compared methods. It verifies that the multiple teacher models originating from different time steps of our post-training procedure can provide more comprehensive knowledge than these jointly finetuned teacher models with different PLMs used in MT-BERT.

4.4 Effectiveness of Multiple Teacher Models

In this subsection, we conduct experiments to explore the impact of the number of teacher models in our Tiny-NewsRec. We vary the number of teacher models M from 1 to 6 and compare the performance of the 4-layer student model on the *Feeds* dataset[4]. The results are shown in Fig. 2. From the results, we find that the performance of

[4] The results on the *MIND* dataset show similar trends and are placed in the Appendix.

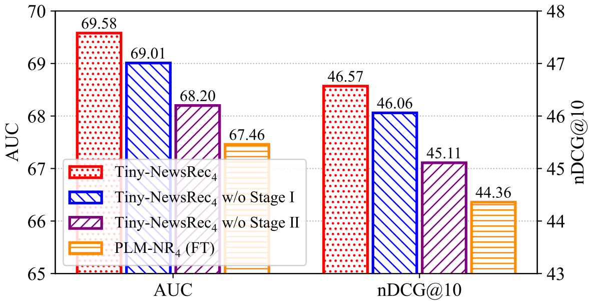

Figure 3: Effectiveness of each stage in our framework.

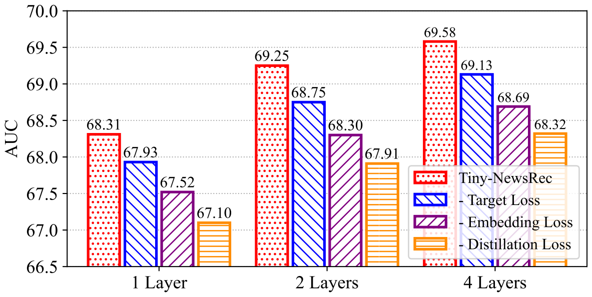

Figure 4: Effectiveness of each loss function.

the student model first greatly improves with the number of teacher models. This is because these teacher models at different time steps of the post-training procedure usually can complement each other. With more teacher models, the student model can receive more comprehensive knowledge and obtain better generalization ability. However, increasing the number of teacher models can only bring marginal improvement when M is larger than 4, which may reach the upper bound of the performance gain brought by the ensemble of multiple teachers. Thus we set M to 4 in our Tiny-NewsRec as a balance between the model performance and the additional training cost of these teacher models.

4.5 Effectiveness of Two-stage Knowledge Distillation

In this subsection, we further conduct several experiments to verify the effectiveness of each stage in our two-stage knowledge distillation method. We compare the performance of the 4-layer student model distilled with our Tiny-NewsRec and its variant with one stage removed on the *Feeds* dataset[4]. The results are shown in Fig. 3. From the results, we first find that the knowledge distillation in Stage II plays a critical role in our approach as the performance of the student model declines significantly when it is removed. This is because the guidance from the teacher models in the second stage such as learned news and user representations can provide much more useful information than the one-hot ground-truth label, which encourages the student model to behave similarly to the teacher models in the news recommendation task. The complement between multiple teacher models also enables the student model to achieve better generalization ability. Second, the performance of the student model also declines after we remove Stage I. This is because our self-supervised domain-specific post-training task can make the PLM better adapt to the news domain and generate more discriminative news representations. The multiple teacher models can also transfer useful domain-specific knowledge to the student model during its domain adaptation.

4.6 Effectiveness of Each Loss Function

In this subsection, we further explore the impact of each part of the overall loss function in our two-stage knowledge distillation method, i.e., the distillation loss ($\mathcal{L}_{\text{DP}}^{\text{distill}}$ and $\mathcal{L}_{\text{FT}}^{\text{distill}}$), the embedding loss ($\mathcal{L}_{\text{DP}}^{\text{emb}}$ and $\mathcal{L}_{\text{FT}}^{\text{emb}}$), and the target loss ($\mathcal{L}_{\text{DP}}^{(s)}$ and $\mathcal{L}_{\text{FT}}^{(s)}$). We compare the performance of the student models distilled with our Tiny-NewsRec approach and its variant with one part of the overall loss function removed. The results on the *Feeds* dataset are shown in Fig. 4. From the results, we have several findings. First, the distillation loss is the most essential part of the overall loss function as the performance of the student model drops significantly after it is removed. This is because the distillation loss can force the student model to make similar predictions as the teacher model, which directly decides the performance of the student model on the news recommendation task. In addition, the embedding loss is also important in our approach. It may be because the embedding loss aligns the news representations and the user representations learned by the student model and the teacher models, which can help the student model better imitate the teacher models. Besides, the target loss is also useful for the training of the student model. This may be because these finetuned teacher models will still make some mistakes in certain training samples. The supervision from the ground-truth label is still necessary for the student model.

4.7 Efficiency Evaluation

In this subsection, we conduct experiments to evaluate the efficiency of the student models distilled with our Tiny-NewsRec. As in news recommendation, encoding news with the PLM-based news encoder is the main computational overhead, we

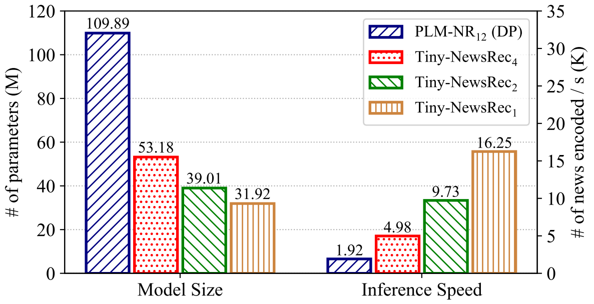

Figure 5: Model size and inference speed of the teacher model and student models.

measure the inference speed of the model in terms of the number of news that can be encoded per second with a single GPU. We also measure the model size by the number of parameters. The evaluation results of the 1-layer, 2-layer, and 4-layer student models and the 12-layer teacher model are shown in Fig. 5. The results show that our Tiny-NewsRec can reduce the model size by 50%-70% and increase the inference speed by 2-8 times while achieving better performance. These results verify that our approach can improve the effectiveness and efficiency of the PLM-based news recommendation model at the same time. It is noted that the student model distilled with other knowledge distillation methods (e.g., TinyBERT and NewsBERT) can achieve the same inference speed as Tiny-NewsRec since the structure of the final student model is the same. However, our Tiny-NewsRec can get much better performance as shown in Table 3 and 4.

5 Conclusion

In this paper, we propose a Tiny-NewsRec method to improve the effectiveness and efficiency of PLM-based news recommendation with domain-specific post-training and a two-stage knowledge distillation method. Specifically, before the task-specific finetuning, we propose to conduct domain-specific post-training towards the PLM-based news encoder with a self-supervised matching task between news titles and news bodies to make the general PLM better capture and match the high-level semantic information in news texts. In our two-stage knowledge distillation method, the student model is first adapted to the news domain and then finetuned on the news recommendation task with the domain-specific and task-specific knowledge transferred from multiple teacher models in each stage. We conduct extensive experiments on two real-world datasets and the results demonstrate that our approach can effectively improve the performance of the PLM-based news recommendation model with considerably smaller models.

6 Ethics Statement

In this paper, we conduct experiments with three real-world datasets, i.e., *MIND*, *Feeds*, and *News*. *MIND* is a public English news recommendation dataset released in (Wu et al., 2020). In this dataset, each user was delinked from the production system when securely hashed into an anonymized ID using one-time salt mapping to protect user privacy. We have agreed to Microsoft Research License Terms[5] before downloading the dataset. *Feeds* is another news recommendation dataset collected on the Microsoft News App. It followed the same processing procedure as *MIND*, using the one-time salt mapping to securely hash each user into an anonymized ID. *News* is a news article dataset collected on the Microsoft News website which only contains public news articles and no user-related information is involved. Thus, all the datasets used in our paper will not reveal any user privacy information.

7 Limitations

In our Tiny-NewsRec, we utilize multiple teacher models to transfer comprehensive knowledge to the student model in our two-stage knowledge distillation method. These teacher models originate from different time steps of the post-training procedure and later they are finetuned with the news recommendation task separately. Our ablation study verifies the effectiveness of multiple teacher models. However, training a teacher model requires lots of time and computing resources as it contains a large PLM. Compared with existing single-teacher knowledge distillation methods, our approach will enlarge the training cost by M times in order to obtain M high-quality teacher models. We will try to reduce the training cost of our approach while keeping its performance in our future work.

8 Acknowledgements

This research was partially supported by grants from the National Key Research and Development Program of China (No. 2021YFF0901003), and the National Natural Science Foundation of China (No. 61922073 and U20A20229).

[5] https://msnews.github.io/

References

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, and Hsiao-Wuen Hon. 2020. UniLMv2: Pseudo-Masked Language Models for Unified Language Model Pre-Training. In *ICML*, pages 642–652. PMLR.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In *EMNLP-IJCNLP*, pages 3615–3620. ACL.

Tuhin Chakrabarty, Christopher Hidey, and Kathy McKeown. 2019. IMHO Fine-Tuning Improves Claim Detection. In *NAACL*, pages 558–563. ACL.

Abhinandan S. Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. 2007. Google News Personalization: Scalable Online Collaborative Filtering. In *WWW*, pages 271–280. ACM.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, pages 4171–4186. ACL.

Kawin Ethayarajh. 2019. How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. In *EMNLP-IJCNLP*, pages 55–65. ACL.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation Degeneration Problem in Training Natural Language Generation Models. In *ICLR*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *ACL*, pages 8342–8360. ACL.

Xiaochuang Han and Jacob Eisenstein. 2019. Unsupervised Domain Adaptation of Contextualized Embeddings for Sequence Labeling. In *EMNLP-IJCNLP*, pages 4238–4248. ACL.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*.

Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *ACL*, pages 328–339. ACL.

Linmei Hu, Chen Li, Chuan Shi, Cheng Yang, and Chao Shao. 2020. Graph Neural News Recommendation with Long-term and Short-term Interest Modeling. *Information Processing & Management*, 57(2):102142.

Qinglin Jia, Jingjie Li, Qi Zhang, Xiuqiang He, and Jieming Zhu. 2021. RMBERT: News Recommendation via Recurrent Reasoning Memory Network over BERT. In *SIGIR*, pages 1773–1777. ACM.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for Natural Language Understanding. In *Findings of EMNLP*, pages 4163–4174. ACL.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

Talia Lavie, Michal Sela, Ilit Oppenheim, Ohad Inbar, and Joachim Meyer. 2010. User Attitudes towards News Content Personalization. *International Journal of Human-Computer Studies*, 68(8):483–495.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: A Pre-trained Biomedical Language Representation Model for Biomedical Text Mining. *Bioinformatics*, 36(4):1234–1240.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020a. On the Sentence Embeddings from Pre-trained Language Models. In *EMNLP*, pages 9119–9130. ACL.

Zhi Li, Bo Wu, Qi Liu, Likang Wu, Hongke Zhao, and Tao Mei. 2020b. Learning the Compositional Visual Coherence for Complementary Recommendations. In *IJCAI*, pages 3536–3543.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Yuang Liu, Wei Zhang, and Jun Wang. 2020. Adaptive Multi-Teacher Multi-level Knowledge Distillation. *Neurocomputing*, 415:106–113.

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-Shot Entity Linking by Reading Entity Descriptions. In *ACL*, pages 3449–3460. ACL.

Wenhao Lu, Jian Jiao, and Ruofei Zhang. 2020. TwinBERT: Distilling Knowledge to Twin-Structured Compressed BERT Models for Large-Scale Retrieval. In *CIKM*, pages 2645–2652. ACM.

Vivek Madan, Ashish Khetan, and Zohar Karnin. 2021. TADPOLE: Task ADapted Pre-Training via AnOmaLy DEtection. In *EMNLP*, pages 5732–5746. ACL.

Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-Based News Recommendation for Millions of Users. In *KDD*, pages 1933–1942. ACM.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. *arXiv preprint arXiv:1807.03748*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient Knowledge Distillation for BERT Model Compression. In *In EMNLP-IJCNLP*, pages 4323–4332. ACL.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices. In *ACL*, pages 2158–2170. ACL.

Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep Knowledge-Aware Network for News Recommendation. In *WWW*, pages 1835–1844.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. In *NeurIPS*, pages 5776–5788.

Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019a. Neural News Recommendation with Attentive Multi-View Learning. In *IJCAI*, pages 3863–3869.

Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019b. NPA: Neural News Recommendation with Personalized Attention. In *KDD*, pages 2576–2584. ACM.

Chuhan Wu, Fangzhao Wu, and Yongfeng Huang. 2021a. One Teacher is Enough? Pre-trained Language Model Distillation from Multiple Teachers. In *Findings of ACL-IJCNLP*, pages 4408–4413. ACL.

Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021b. Empowering News Recommendation with Pre-Trained Language Models. In *SIGIR*, pages 1652–1656. ACM.

Chuhan Wu, Fangzhao Wu, Yang Yu, Tao Qi, Yongfeng Huang, and Qi Liu. 2021c. NewsBERT: Distilling Pre-trained Language Model for Intelligent News Application. In *Findings of EMNLP*, pages 3285–3295. ACL.

Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. 2020. MIND: A Large-scale Dataset for News Recommendation. In *ACL*, pages 3597–3606. ACL.

Canwen Xu, Wangchunshu Zhou, Tao Ge, Furu Wei, and Ming Zhou. 2020. BERT-of-Theseus: Compressing BERT by Progressive Module Replacing. In *EMNLP*, pages 7859–7869. ACL.

Qi Zhang, Qinglin Jia, Chuyuan Wang, Jingjie Li, Zhaowei Wang, and Xiuqiang He. 2021a. AMM: Attentive Multi-Field Matching for News Recommendation. In *SIGIR*, pages 1588–1592. ACM.

Qi Zhang, Jingjie Li, Qinglin Jia, Chuyuan Wang, Jieming Zhu, Zhaowei Wang, and Xiuqiang He. 2021b. UNBERT: User-News Matching BERT for News Recommendation. In *IJCAI*, pages 3356–3362.

Qiannan Zhu, Xiaofei Zhou, Zeliang Song, Jianlong Tan, and Li Guo. 2019. DAN: Deep Attention Neural Network for News Recommendation. In *AAAI*, pages 5973–5980.

A Appendix

A.1 Experimental Settings

In our domain-specific post-training, we use the first 24 tokens of the news title and the first 512 tokens of the news body for news title and news body modeling. We use the pre-trained UniLMv2 model as the PLM and only finetune its last three Transformer layers. During finetuning with the news recommendation task, we use the first 30 tokens of the news title for news modeling. We also use the UniLMv2 model as the PLM and only finetune its last two Transformer layers as we find that finetuning all the parameters does not bring significant gain in model performance but drastically slows down the training speed. The complete hyper-parameter settings are listed in Table 5.

A.2 Additional Results on *MIND*

We also report the additional results on the *MIND* dataset, which are shown in Fig. 6 and Fig. 7. We observe phenomena similar to the results on the *Feeds* dataset.

A.3 Experimental Environment

We conduct experiments on a Linux server with Ubuntu 18.04.1. The server has 4 Tesla V100-SXM2-32GB GPUs with CUDA 11.0. The CPU is Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz and the total memory is 661GB. We use Python 3.6.9 and PyTorch 1.6.0. In our domain-specific post-training and the post-training stage knowledge distillation experiments, the model is trained on a single GPU. All the other models are parallelly trained on 4 GPUs with the Horovod framework.

A.4 Running Time

On the *News* dataset, the domain-specific post-training of the 12-layer teacher model and the post-training stage knowledge distillation of the 4-layer, 2-layer, and 1-layer student models takes around 12 hours, 10 hours, 8 hours, and 6 hours respectively with a single GPU. On the *MIND* dataset, the fine-tuning of the 12-layer teacher model and the finetuning stage knowledge distillation of the 4-layer, 2-layer, and 1-layer student models takes around 12 hours, 10 hours, 8 hours, and 6 hours respectively with 4 GPUs, while on the *Feeds* dataset, it takes 3 hours, 2.5 hours, 2 hours, and 1.5 hours respectively.

General Hyper-parameters	
Dimension of query vector in attention network	200
Adam betas	(0.9, 0.999)
Adam eps	1e-8
Domain-specific Post-training	
Negative sampling ratio N	9
Dimension of news title/body representation	256
Batch size	32
Learning rate	1e-6
News Recommendation Finetuning	
Negative sampling ratio S	4
Max number of historical clicked news L	50
Dimension of news/user representation	256
Batch size	32×4
Learning rate	5e-5
Two-stage Knowledge Distillation	
Temperature T_{DP}	1
Temperature T_{FT}	1
Number of teacher models M	4

Table 5: Hyper-parameter settings

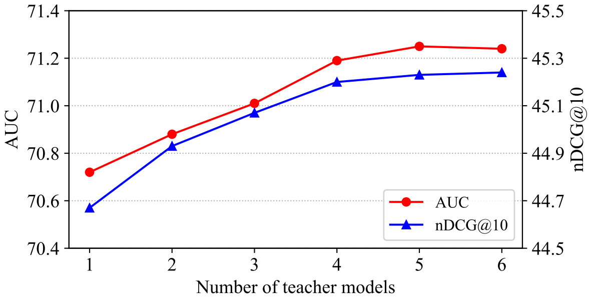

Figure 6: Impact of the number of teacher models M.

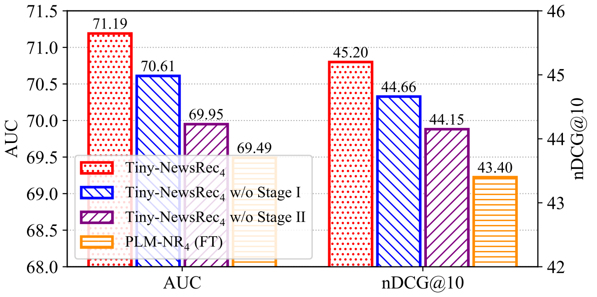

Figure 7: Effectiveness of each stage in our framework.