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

Yang Yu[1], **Fangzhao Wu**[2], **Chuhan Wu**[3], **Jingwei Yi**[1], **Tao Qi**[3], **Qi Liu**[1]
[1]University of Science and Technology of China, Hefei 230027, China
[2]Microsoft Research Asia, Beijing 100080, China
[3]Department of Electronic Engineering & BNRist, Tsinghua University, Beijing 100084, China
yflyl613@mail.ustc.edu.cn, {wufangzhao, wuchuhan15}@gmail.com
yjw1029@mail.ustc.edu.cn, taoqi.qt@gmail.com, qiliuql@ustc.edu.cn

Abstract

Personalized news recommendation has been widely adopted to improve user experience. Recently, pre-trained language models (PLMs) have demonstrated the great capability of natural language understanding and the potential of improving news modeling for news recommendation. However, existing PLMs are usually pre-trained on general corpus such as BookCorpus and Wikipedia, which have some gaps with the news domain. Directly finetuning PLMs with the news recommendation task may be sub-optimal for news understanding. Besides, PLMs usually contain a large volume of parameters and have high computational overhead, which imposes a great burden on the 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. In order to reduce the domain gap between general corpora and the news data, we propose a self-supervised domain-specific post-training method to adapt the generally pre-trained language models to the news domain with the task of news title and news body matching. To improve the efficiency of PLM-based news recommendation while maintaining the performance, we propose a two-stage knowledge distillation method. In the first stage, we use the domain-specific teacher PLM to guide the student model for news semantic modeling. In the second stage, we use a multi-teacher knowledge distillation framework to transfer the comprehensive knowledge from a set of teacher models finetuned for news recommendation to the student. Experiments on two real-world datasets show that our methods can achieve better performance in news recommendation with smaller models.

Introduction

With the explosion of information, massive news are 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 news they are interested in (Okura et al. 2017). To tackle this problem, many news recommendation methods have been proposed to provide personalized news feeds and alleviate information overload 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 accurate news representations from news texts is the prerequisite for high-quality news recommendation (Wu et al. 2020). Most existing news recommendation methods use shallow NLP models to learn news representations. For example, An et al. (2019) propose to use a CNN network to learn contextual word representations by capturing local context information and use an attention network to select important words in news titles. Wu et al. (2019c) propose to use a multi-head self-attention layer to capture the global relation between words in news titles, and also use an attention network to compute the news representation. However, it is difficult for these shallow models to accurately capture the deep semantic information in news texts (Devlin et al. 2019), which limits their performance on news recommendation.

Pre-trained language models (PLMs) are 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 news understanding in news recommendation (Wu et al. 2021a; Xiao et al. 2021; Zhang et al. 2021). For example, Wu et al. (2021a) propose to replace these shallow models with the PLM to capture the deep contexts in news texts. However, these methods simply finetune the PLM with the news recommendation task, the supervision from which may not optimally train the PLM to capture semantic information in news texts and may be insufficient to solve the domain shift problem. Moreover, PLMs usually have a large number of parameters (Lan et al. 2020). For example, the BERT-base model (Devlin et al. 2019) contains 12 Transformer layers (Vaswani et al. 2017) and up to 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 design a self-supervised domain-specific post-training method to adapt the generally pre-trained language models to the news domain with the task of news title and news body matching. In this way, the domain-specific PLM-based news encoder can better capture the semantic information in news texts and generate more discriminative representations, which are beneficial for news content understanding and user inter-

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

est matching in the following news recommendation task. In addition, we propose a two-stage knowledge distillation method to compress the large PLM while maintaining its performance[2]. In the first stage, the student PLM is forced to mimic the domain-specifically post-trained teacher PLM in the matching task between news titles and news bodies to learn news semantic modeling. In the second stage, the domain-specific teacher PLM is first finetuned with different random seeds on the news recommendation task to obtain multiple task-specific teachers. Then we propose a multi-teacher knowledge distillation framework to transfer task-specific knowledge from these teacher models to the student model. Since different teachers may have different abilities on different samples, for each training sample, we assign different teachers with different weights based on their performance on this sample, which allows the student model to 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 this paper are as follows:

- We propose a Tiny-NewsRec approach to improve both the effectiveness and the efficiency of PLM-based news recommendation.
- We propose to domain-specifically post-train the PLM-based news encoder with a self-supervised matching task between news titles and news bodies before task-specific finetuning to better fill the domain gap.
- We propose a two-stage knowledge distillation method with multiple teacher models to compress the large PLM-based news recommendation model while maintaining its performance.
- 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.

Related Work

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 (Xiao et al. 2021; Zhang et al. 2021; Wu et al. 2021a). For example, Zhang et al. (2021) proposed a UNBERT approach, which is a BERT-based user-news matching model. It takes in the concatenation of the user's historical clicked news and the candidate news, and uses the PLM to capture multi-grained user-news matching signals at both word-level and news-level. Wu et al. (2021a) 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 general corpora and the news domain (Gururangan et al. 2020). Besides, the PLMs are usually with large parameter sizes and high computational overhead (Lan et al. 2020). Different from these methods, our approach can better fill the domain gap with an additional domain-specific post-training task and further reduce the computational cost with the two-stage knowledge distillation method.

PLM Knowledge Distillation

Knowledge distillation is a technique that aims to compress a heavy teacher model into a lightweight student model while maintaining its performance (Hinton, Vinyals, and Dean 2015). In recent years, many works explore to compress large-scale PLMs via knowledge distillation (Tang et al. 2019a; Mirzadeh et al. 2020; Wang et al. 2020; Sun et al. 2020; Xu et al. 2020). For example, Tang et al. (2019b) utilized the output soft label of a BERT-large model to distill it into a single-layer BiLSTM. Sun et al. (2019b) proposed a patient knowledge distillation approach named BERT-PKD, which lets the student model learn from both the output soft labels from the last layer of the teacher model and the hidden states produced by intermediate layers. Sanh et al. (2020) proposed a DistilBERT approach, which distills the student model at the pre-training stage with a combination of language modeling, distillation, and embedding cosine-distance losses. Jiao et al. (2020) proposed a TinyBERT approach, which lets the student model imitate the output probabilities, embeddings, hidden states, and attention score matrices of the teacher model at both the pre-training stage and the finetuning stage. There are also a few works that aim to distill the PLM for specific downstream tasks (Lu, Jiao, and Zhang 2020; Wu et al. 2021b). For example, Wu et al. (2021b) proposed a NewsBERT approach for intelligent news applications. A teacher-student joint learning and distillation framework is proposed to collaboratively learn both teacher and student models. A momentum distillation method is also designed to incorporate the gradients of the teacher model into the update of the student model which can better transfer the useful knowledge learned by the teacher model. However, these knowledge distillation methods neglect the potential domain gap between the pre-training corpora and the downstream task domain. Besides, they only use one teacher model to guide the training of the student model, which may provide insufficient or even biased supervision (Wu, Wu, and Huang 2021). Therefore, we first use a large domain-specifically post-trained teacher model to help the student model better adapt to the news domain. Then we use a multi-teacher knowledge distillation framework to transfer richer knowledge from a set of finetuned teacher models to the student.

Methodology

In this section, we introduce the details of our Tiny-NewsRec approach, which can fill the domain gap between general corpora and the news domain, and distill the large PLM for news recommendation applications. We first introduce the structure of the PLM-based news recommendation model. Then we introduce the self-supervised matching

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

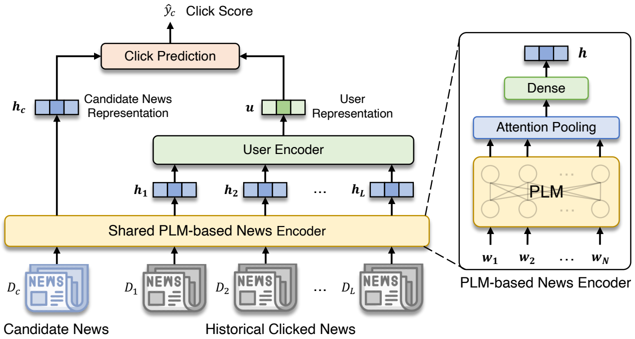


Figure 1: Structure of our PLM-based news recommendation model.

task between news titles and news bodies used for domain-specific post-training. Finally, we introduce the workflow of our two-stage knowledge distillation method with a multi-teacher knowledge distillation framework.

News Recommendation Model

We first introduce the overall structure of our PLM-based news recommendation model. As shown in Fig.1, it consists of three major components, i.e. a shared PLM-based news encoder, a user encoder, and a click prediction module. The shared news encoder aims to learn news representations from news texts. Following PLM-NR (Wu et al. 2021a), we use a PLM to get the contextual representation of each token in the input news. Then we use an attention network to aggregate these representations and feed its output into a dense layer to get the final news representation. The user encoder aims to learn the user representation $\mathbf{u}$ from the representations of the last L news clicked by the user, i.e. $[\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_L]$. Following Wu et al. (2019a), we implement it with an attention network to select important news. In the click prediction module, we use dot product to calculate the relevance score between the candidate news representation $\mathbf{h}_c$ and the target user representation $\mathbf{u}$, and take it as the predicted result, i.e. $\hat{y}_c = \mathbf{h}_c^T \mathbf{u}$.

Domain-specific Post-training

Since the supervision from the news recommendation task may not optimally train the PLM to understand the news content, directly finetuning the PLM on the news recommendation data may be insufficient to fill the domain gap between general corpora and the news corpus (Gururangan et al. 2020). In order to better adapt the PLM to the news domain, we propose to conduct domain-specific post-training before finetuning it with the news recommendation task. We design a self-supervised matching task between news titles and news bodies which can make the PLM-based news encoder better at capturing and matching semantic information in news texts. Given a pair of news title and news body, the news encoder is trained to predict whether they come from the same news article. The model structure for this task is shown in the right half of Fig.2(a). The architecture of the PLM-based news encoder is the same as that in Fig.1.

Following previous works (Huang et al. 2013; Wu et al. 2019a), we adopt the negative sampling method to construct the training samples. Given the i-th news body, we take its corresponding news title as the positive sample and randomly select N other news titles as negative samples. We use the PLM-based news encoder to get the news body representation $\mathbf{h}_b$ and the news title representations $\mathbf{h}_t = [\mathbf{h}_t^+, \mathbf{h}_{t_1}^-, ..., \mathbf{h}_{t_N}^-]$. Then we take the dot product of the news body representation and each news title representation as the predicted score, which is denoted as $[\hat{y}^+, \hat{y}_1^-, ...\hat{y}_N^-]$. These predicted scores are further normalized with the softmax function and the predicted probability of the positive sample is formulated as follows:

$$p_i = \frac{\exp\left(\hat{y}^+\right)}{\exp\left(\hat{y}^+\right) + \sum_{j=1}^{N} \exp\left(\hat{y}_j^-\right)}.$$

To maximize the predicted probability of the positive sample, we use the Cross-Entropy loss as the loss function, which can be formulated as follows:

$$\mathcal{L}_{match} = -\sum_{i \in \mathcal{T}} \log(p_i),$$

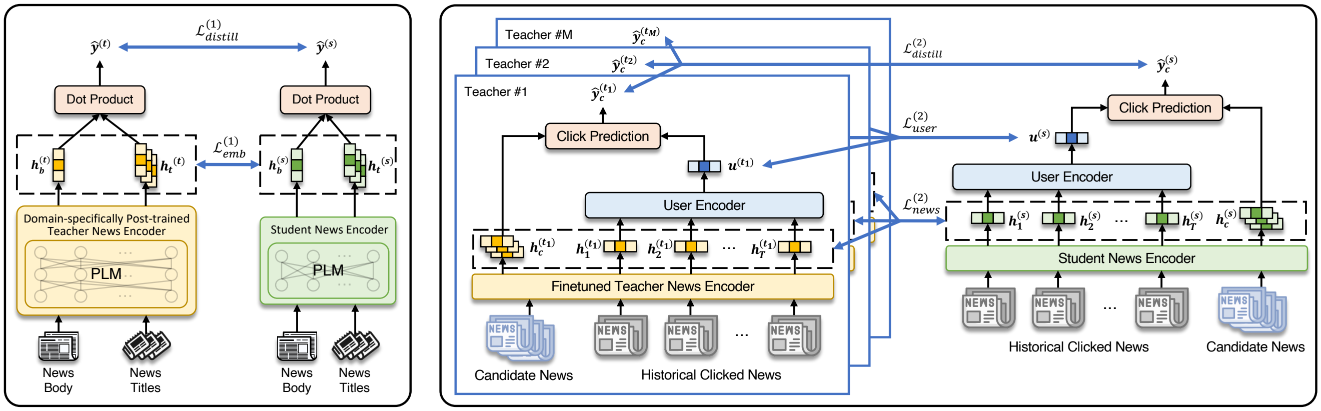
where $\mathcal{T}$ is the set of positive training samples.

In this way, the domain-specifically post-trained PLM-based news encoder can generate more similar representations for related texts and distinguish them from the others, which can alleviate the anisotropy problem of the sentence embedding generated by the PLM (Gao et al. 2019; Ethayarajh 2019; Li et al. 2020). As a result, the news representations generated by the news encoder can be more discriminative and better at capturing semantic similarity, which is beneficial to the user interests matching in the following news recommendation task.

Two-stage Knowledge Distillation

Although with our proposed domain-specifically post-train then finetune procedure, the PLM-based news recommendation model can achieve superior performance, it still has high computational overhead and is difficult to meet the speed requirement of low-latency online services. In order to achieve our goal of efficiency, we further propose a two-stage knowledge distillation method. The overall framework is shown in Fig.2. In our method, the student model is first trained to imitate the domain-specifically post-trained teacher model in the matching task between news titles and news bodies. Then we finetune the domain-specifically post-trained teacher model with different random seeds on the news recommendation task and use these finetuned teacher models to guide the finetuning of the student model via a multi-teacher knowledge distillation framework.

In the first stage, in order to help the student PLM better adapt to the news domain, we use the large domain-specifically post-trained teacher news encoder to guide the student model in the matching task. The model framework is shown in Fig.2(a). To encourage the student model to make similar predictions as the teacher model in the matching task, we use a distillation loss to force the student model to imitate the output soft labels of the teacher model. Given



(a) First stage knowledge distillation.

(b) Second stage knowledge distillation.

Figure 2: Illustration of our two-stage knowledge distillation method.

a piece of news body and $N+1$ news titles, the soft labels predicted by the teacher model and the student model are denoted as $\hat{\mathbf{y}}^{(t)} = [\hat{y}^{+(t)}, \hat{y}_1^{-(t)}, ..., \hat{y}_N^{-(t)}]$ and $\hat{\mathbf{y}}^{(s)} = [\hat{y}^{+(s)}, \hat{y}_1^{-(s)}, ..., \hat{y}_N^{-(s)}]$ respectively. The distillation loss in the first stage is formulated as follows:

$$\mathcal{L}^{(1)}_{distill} = T_1^2 \cdot \mathrm{CE}(\hat{\mathbf{y}}^{(t)}/T_1, \hat{\mathbf{y}}^{(s)}/T_1),$$

where T_1 is the temperature hyper-parameter in the first stage that controls the smoothness of the predicted probability distribution of the teacher model, and CE stands for the Cross-Entropy loss function. Besides, the learned news title representations and news body representations are very important in the matching task, which will directly affect the final predicted score. Therefore we use an embedding loss to align the output representations of the teacher news encoder and the student news encoder. Denote the news title representations and the news body representation learned by the teacher news encoder as $\mathbf{h}_t^{(t)} = [\mathbf{h}_t^{+(t)}, \mathbf{h}_{t_1}^{-(t)}, ..., \mathbf{h}_{t_N}^{-(t)}]$ and $\mathbf{h}_b^{(t)}$, and denote these representations learned by the student news encoder as $\mathbf{h}_t^{(s)} = [\mathbf{h}_t^{+(s)}, \mathbf{h}_{t_1}^{-(s)}, ..., \mathbf{h}_{t_N}^{-(s)}]$ and $\mathbf{h}_b^{(s)}$ respectively, the embedding loss in the first stage is formulated as follows:

$$\mathcal{L}^{(1)}_{emb} = \mathrm{MSE}(\mathbf{h}_t^{(t)}, \mathbf{h}_t^{(s)}) + \mathrm{MSE}(\mathbf{h}_b^{(t)}, \mathbf{h}_b^{(s)}),$$

where MSE stands for the Mean Squared Error loss function. The overall loss function for the student model in the first stage knowledge distillation is the weighted summation of the distillation loss, the embedding loss, and the target loss, which is formulated as follows:

$$\mathcal{L}^{(1)} = \mathcal{L}^{(1)}_{distill} + \mathcal{L}^{(1)}_{emb} + \beta_1 \cdot \mathrm{CE}(\hat{\mathbf{y}}^{(s)}, \mathbf{y}),$$

where $\mathbf{y}$ is the one-hot ground-truth label of the matching task and β_1 is the hyper-parameter that controls the impact of the teacher model in the first stage knowledge distillation.

In the second stage, in order to transfer more comprehensive knowledge to the student model during finetuning with the news recommendation task, we propose a multi-teacher knowledge distillation framework which is shown in Fig.2(b). We first finetune the domain-specifically post-trained teacher model with M different random seeds on the news recommendation task. Then these M teacher models are used to guide the finetuning of the student news encoder that got from the first stage knowledge distillation. For each training sample, we assign a weight to the i-th teacher model according to its performance on this sample, which is measured by the Cross-Entropy loss between its predicted score $\hat{\mathbf{y}}_c^{(t_i)}$ and the ground-truth label of the input training sample $\mathbf{y}_c$. The loss is further multiplied with a positive learnable parameter ω which is used to enlarge the difference between teacher models. Denote the weight of the i-th teacher model on a training sample as w_i, it is formulated as follows:

$$w_i = \frac{\exp(-\mathrm{CE}(\hat{\mathbf{y}}_c^{(t_i)}, \mathbf{y}_c) \times \omega)}{\sum_{j=1}^{M} \exp(-\mathrm{CE}(\hat{\mathbf{y}}_c^{(t_j)}, \mathbf{y}_c) \times \omega)},$$

Similar to the first stage, we use the distillation loss to force the student model to make similar predictions as the best teacher model on a training sample. Since now we have several teacher models with different weights, we use the weighted summation of all the soft labels of teacher models as guidance. Therefore the distillation loss is formulated as follows:

$$\mathcal{L}^{(2)}_{distill} = T_2^2 \cdot \mathrm{CE}(\sum_{i=1}^{M} w_i \cdot \hat{\mathbf{y}}_c^{(t_i)}/T_2, \hat{\mathbf{y}}_c^{(s)}/T_2).$$

where T_2 is the temperature hyper-parameter in the second stage. In addition, since the news representation and the user representation are the keys in the news recommendation task, we also let the student model imitate the learned news representations and user representations of teacher models. Considering that the representations learned by each teacher model may lie in different spaces, we use an additional dense layer for each teacher model to project their learned representations into one unified space. The embedding loss of news representations and user representations between the i-th teacher model and the student model are denoted as $\mathcal{L}^{(2)}_{news_i}$ and $\mathcal{L}^{(2)}_{user_i}$ respectively, which are formulated as follows:

$$\mathcal{L}^{(2)}_{news_i} = \text{MSE}(\mathbf{W}^{(t_i)}_{news} \times \mathbf{h}^{(t_i)}_{news} + \mathbf{b}^{(t_i)}_{news}, \mathbf{h}^{(s)}_{news}),$$
$$\mathcal{L}^{(2)}_{user_i} = \text{MSE}(\mathbf{W}^{(t_i)}_{user} \times \mathbf{u}^{(t_i)} + \mathbf{b}^{(t_i)}_{user}, \mathbf{u}^{(s)}),$$

where $\mathbf{h}^{(t_i)}_{news}$ and $\mathbf{h}^{(s)}_{news}$ represent the news representations of the input historical clicked news and the candidate news learned by the i-th teacher model and the student model respectively. $\mathbf{W}^{(t_i)}_{news}$, $\mathbf{b}^{(t_i)}_{news}$ and $\mathbf{W}^{(t_i)}_{user}$, $\mathbf{b}^{(t_i)}_{user}$ are the learnable parameters used to project the news representations and user representations learned by the i-th teacher model. The total embedding loss is the weighted summation of all the embedding losses of news representations and user representations between the student model and each teacher model, i.e. $\mathcal{L}^{(2)}_{emb} = \sum_{i=1}^{M} w_i \cdot (\mathcal{L}^{(2)}_{news_i} + \mathcal{L}^{(2)}_{user_i})$. The overall loss function for the student model in the second stage knowledge distillation is also the weighted summation of the distillation loss, the embedding loss, and the target loss, which is formulated as follows:

$$\mathcal{L}^{(2)} = \mathcal{L}^{(2)}_{distill} + \mathcal{L}^{(2)}_{emb} + \beta_2 \cdot \text{CE}(\hat{\mathbf{y}}^{(s)}_c, \mathbf{y}_c),$$

where β_2 controls the impact of the teacher models in the second stage knowledge distillation[3].

Experiments

Datasets and Experiment Settings

We conduct experiments with three real-world datasets, i.e. *MIND*, *Feeds* and *News*. *MIND*[4] 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[5]. *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 sample 20% impressions from the training set for validation. *News* contains news articles collected on the Microsoft News website from 2020-08-01 to 2020-09-20, which is used for domain-specific post-training. Detailed statistics of these datasets are summarized in Table 1.

In our experiments, we apply the pre-trained UniLMv2 (Bao et al. 2020) to initialize the PLM in the news encoder. The dimension of the news representation and the query vector in the attention network is 256 and 200 respectively. The temperature hyper-parameters T_1 and T_2 are both set to 1. β_1 and β_2 in the loss functions of our two-stage knowledge distillation method are set to 1 and 0.1 respectively. The number of teacher models M is set to 4. We use the Adam optimizer (Kingma and Ba 2015) for model training. The detailed experiment settings are listed in Appendix. All the hyper-parameters are tuned on the validation set. Following Wu et al. (2020), we use AUC, MRR, nDCG@5, 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.

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 1: Detailed statistics of *MIND*, *Feeds* and *News*.

Performance Comparison

In this section, we compare the performance of the teacher model PLM-NR-12 (**D**omain-specific **P**re-train) that trained with our domain-specific post-train then finetune procedure, and the student models trained with our Tiny-NewsRec approach with several baseline methods, including:

- **PLM-NR** (**F**ine**t**une) (Wu et al. 2021a), a method which applies the PLM in the news encoder and directly finetunes it with the news recommendation task. We compare the performance of its 12-layer UniLMv2 version and its variant using the first 1, 2, or 4 layers.
- **PLM-NR** (**F**urther **P**re-train) (Sun et al. 2019a), a variant of PLM-NR where we first further pre-train the UniLMv2 model with the MLM task (Devlin et al. 2019) on the *News* dataset and then finetune 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. For a fair comparison, we compare the performance of the 1-layer, 2-layer, and 4-layer student models distilled from the PLM-NR-12 (DP).
- **NewsBERT** (Wu et al. 2021b), a PLM knowledge distillation method specialized for intelligent news applications. For a fair comparison, we use the domain-specifically post-trained 12-layer UniLMv2 model to initialize the PLM in the teacher model and jointly train it with the 1-layer, 2-layer, or 4-layer student model.

Table 2 shows the performance of all the compared methods on the *MIND* and *Feeds* datasets. From the results, we have the following observations. First, comparing 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. This is because in the first stage the domain-specifically post-trained teacher model can help the student fill the domain gap between general corpora and the news domain. Besides, we use a multi-teacher knowledge distillation framework which can transfer richer knowledge to the student model. Second, our Tiny-NewsRec achieves comparable performance with the teacher model PLM-NR-12 (DP). It

[3] The effectiveness of each part of the loss function is verified in Appendix.

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

[5] We randomly choose 1/4 samples from the training set as our training data due to the limitation of training speed.

Model	MIND				Feeds			
	AUC	MRR	nDCG@5	nDCG@10	AUC	MRR	nDCG@5	nDCG@10
PLM-NR-12 (FT)	69.72±0.15	34.74±0.10	37.99±0.11	43.71±0.07	67.93±0.13	34.42±0.07	37.46±0.09	45.09±0.07
PLM-NR-12 (FP)	69.82±0.14	34.90±0.11	38.17±0.09	43.83±0.07	68.11±0.11	34.49±0.12	37.58±0.07	45.11±0.08
PLM-NR-12 (DP)*	**70.20±0.10**	**35.27±0.08**	**38.54±0.07**	**44.20±0.08**	**68.71±0.08**	**35.10±0.09**	**38.32±0.06**	**45.83±0.08**
PLM-NR-4 (FT)	69.49±0.14	34.40±0.10	37.64±0.10	43.40±0.09	67.46±0.12	33.71±0.11	36.69±0.08	44.36±0.09
PLM-NR-2 (FT)	68.99±0.08	33.59±0.14	36.81±0.11	42.61±0.11	67.05±0.14	33.33±0.09	36.15±0.10	43.90±0.12
PLM-NR-1 (FT)	68.12±0.12	33.20±0.07	36.29±0.09	42.07±0.10	66.26±0.10	32.55±0.12	35.22±0.07	42.99±0.09
TinyBERT-4	69.77±0.13	34.83±0.09	38.02±0.11	43.69±0.09	67.73±0.11	34.00±0.08	37.03±0.10	44.59±0.12
TinyBERT-2	69.44±0.17	34.11±0.07	37.55±0.08	43.14±0.07	67.35±0.13	33.69±0.05	36.59±0.08	44.21±0.09
TinyBERT-1	68.42±0.12	33.55±0.10	36.69±0.09	42.35±0.08	66.53±0.10	32.81±0.07	35.61±0.11	43.29±0.09
NewsBERT-4	69.85±0.17	34.91±0.09	38.19±0.09	43.84±0.08	68.34±0.13	34.58±0.06	37.69±0.09	45.27±0.08
NewsBERT-2	69.62±0.09	34.67±0.12	37.86±0.11	43.54±0.11	67.90±0.07	34.26±0.09	37.29±0.10	44.86±0.11
NewsBERT-1	68.67±0.11	33.95±0.07	37.05±0.14	42.74±0.13	67.00±0.10	33.24±0.11	36.09±0.08	43.80±0.07
Tiny-NewsRec-4*	**70.40±0.05**	**35.43±0.08**	**38.76±0.05**	**44.43±0.04**	**68.93±0.06**	**35.21±0.09**	**38.43±0.08**	**45.97±0.10**
Tiny-NewsRec-2	70.28±0.07	35.32±0.07	38.65±0.07	44.28±0.08	68.58±0.03	34.82±0.07	38.02±0.09	45.57±0.07
Tiny-NewsRec-1	69.85±0.03	34.93±0.08	38.21±0.09	43.84±0.09	68.14±0.05	34.53±0.07	37.61±0.08	45.14±0.08

Table 2: Performance comparisons of different models. (FT=Finetune, FP=Further Pre-train, DP=Domain-specific Post-train) *Improvements over other baselines are significant at $p < 0.01$ (by comparing the models with the same number of layers).

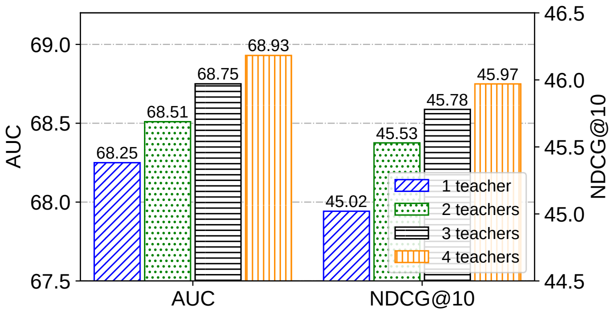


Figure 3: Impact of different number of teacher models.

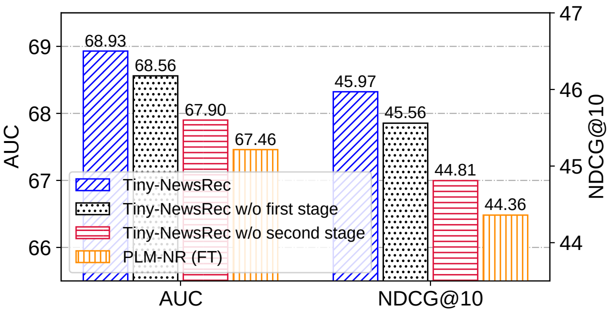


Figure 4: Effectiveness of each stage in our framework.

is noted that Tiny-NewsRec contains much fewer parameters and has lower computational overhead, which validates the effectiveness of our two-stage knowledge distillation method. Third, with the same parameter size, methods applying knowledge distillation (i.e. TinyBERT, NewsBERT, and Tiny-NewsRec) outperform the traditional pretrain then finetune paradigm (i.e. PLM-NR (FT)). This is because the guidance from the teacher model such as output soft labels can provide more useful information than the ground-truth label. Fourth, PLM-NR-12 (FP) outperforms PLM-NR-12 (FT). This is because further pre-training the UniLMv2 model can make it specialized to the news data distribution, therefore boost its ability of news modeling. Finally, PLM-NR-12 (DP) outperforms PLM-NR-12 (FP). This is because our proposed matching task can help the PLM-based news encoder better at capturing the semantic information in news texts and generate more discriminative news representations, which can effectively help the user interest matching in the following news recommendation task.

Effectiveness of Multiple Teacher Models

In this section, we conduct experiments to validate the effectiveness of using multiple teacher models in our second stage knowledge distillation. We vary the number of teacher models M from 1 to 4 and compare the performance of the 4-layer student model on the *Feeds* dataset[6]. The results are shown in Fig.3. From the results, we find that the performance of the student model improves with the number of teacher models. This may be because these teacher models usually can complement each other. With more teacher models, the student model can receive more comprehensive knowledge and obtain better generalization ability.

Effectiveness of Two-stage Knowledge Distillation

In this section, 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 approach and its variant with one stage removed on the *Feeds* dataset. The results are shown in Fig.4. From the results, we first find that the second stage knowledge distillation 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 multiple teacher models in the second stage such as learned news and

[6] We only include results on the *Feeds* dataset due to space limit. The results on the *MIND* dataset are in Appendix.

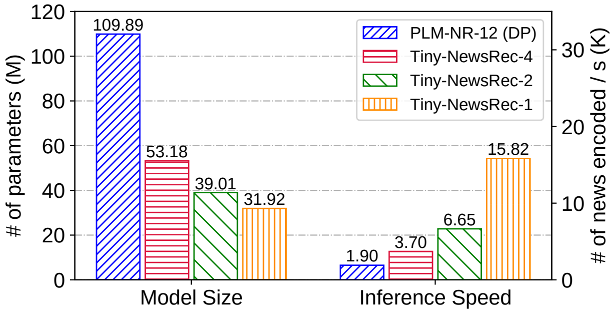


Figure 5: Model size and inference speed of the teacher model and student models.

user representations can provide much more useful information than the ground-truth label, which encourages the student model to behave similarly as the teacher models. The complement between these teacher models also enables the student model to have better generalization ability. Second, the performance of the student model also declines after we remove the first stage knowledge distillation, and the student model that only learns from the domain-specifically post-trained teacher model in the first stage and finetunes on news recommendation data alone still outperforms PLM-NR (FT). This is because our matching task used for domain-specific post-training can better adapt the PLM to the news domain and enable it to generate more discriminative news representations, which can be transferred to the following news recommendation task and boosts the performance of the PLM-based news recommendation model.

Efficiency Evaluation

In this section, we conduct experiments to evaluate the efficiency of the student models distilled with our Tiny-NewsRec approach. As in news recommendation, encoding news with the PLM-based news encoder is the main computational overhead, we measure the inference speed of the model in terms of the number of news that can be encoded per second with a single GPU. The test results and the number of parameters of the 1-layer, 2-layer, and 4-layer student models and the 12-layer teacher model PLM-NR-12 (DP) are shown in Fig.5. The results show that our Tiny-NewsRec method can reduce the model size by 50%-70% and increase the inference speed by 2-8 times while achieving comparable or even 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.

Hyper-parameter Analysis

As shown in Fig. 6, we analyze the influence of two important hyper-parameters, β_1 and β_2 in the loss functions of our two-stage knowledge distillation method on the *Feeds* dataset. First, we fix β_1 to 1.0 and vary the value of β_2 from 0 to 0.3. We find that the performance is not optimal when β_2 is close to 0, and a relatively large β_2 (e.g. $\beta_2 > 0.15$) also hurts the performance. This may be because in the second stage knowledge distillation, the supervision signals from both finetuned teacher models and ground-truth labels are useful, while those from teacher models are more important. Thus, a moderate selection of β_2 from 0.05 to 0.15 is recommended. Then we fix β_2 to 0.1 and vary the value of β_1 from 0.7 to 1.3. We find that the model achieves optimal performance when β_1 is set to 1, and the performance declines when we either increase or decrease the value of β_1. This may be because in the first stage knowledge distillation we only use one domain-specifically post-trained teacher model to guide the student model. The supervision from the teacher model and the ground-truth label may have equal contributions and complement each other. Thus, setting the value of β_1 around 1 is recommended.

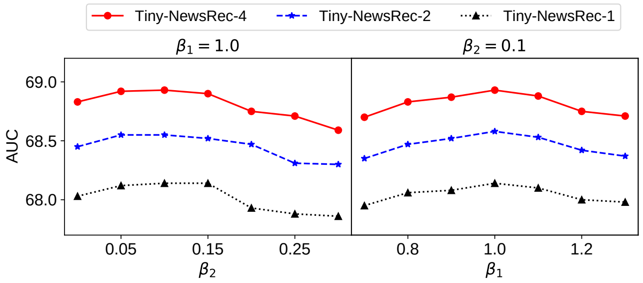


Figure 6: Influence of the hyper-parameter β_1 and β_2.

Conclusion and Future Work

In this paper, we propose a Tiny-NewsRec approach to improve the effectiveness and the efficiency of PLM-based news recommendation with domain-specific post-training and a two-stage knowledge distillation method. Before finetuning, we conduct domain-specific post-training on the PLM-based news encoder with a self-supervised matching task between news titles and news bodies to make the generally pre-trained PLM better model the semantic information in news texts. In our two-stage knowledge distillation method, the student model can first adapt to the news domain with the guidance from the domain-specifically post-trained teacher model. Then a multi-teacher knowledge distillation framework is used to transfer task-specific knowledge from a set of finetuned teacher models to the student during finetuning. 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.

In the future, we plan to deploy our Tiny-NewsRec in online personalized news recommendation service to verify its online performance. Besides, comparing with these single-teacher knowledge distillation methods, our approach will introduce additional training cost in order to train multiple teacher models. We are also interested in reducing the training cost of our two-stage knowledge distillation method while keeping its performance.

References

An, M.; Wu, F.; Wu, C.; Zhang, K.; Liu, Z.; and Xie, X. 2019. Neural News Recommendation with Long- and Short-term User Representations. In *ACL*, 336–345.

Bao, H.; Dong, L.; Wei, F.; Wang, W.; Yang, N.; Liu, X.; Wang, Y.; Piao, S.; Gao, J.; Zhou, M.; and Hon, H.-W. 2020. UniLMv2: Pseudo-Masked Language Models for Unified Language Model Pre-Training. In *ICML*, 642–652.

Das, A. S.; Datar, M.; Garg, A.; and Rajaram, S. 2007. Google news personalization: scalable online collaborative filtering. In *WWW*, 271–280.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*, 4171–4186.

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

Gao, J.; He, D.; Tan, X.; Qin, T.; Wang, L.; and Liu, T.-Y. 2019. Representation Degeneration Problem in Training Natural Language Generation Models. In *ICLR*.

Gururangan, S.; Marasović, A.; Swayamdipta, S.; Lo, K.; Beltagy, I.; Downey, D.; and Smith, N. A. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *ACL*, 8342–8360.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*.

Hu, L.; Li, C.; Shi, C.; Yang, C.; and Shao, C. 2020. Graph Neural News Recommendation with Long-term and Short-term Interest Modeling. *Inf. Process. Manag.*, 57(2): 102142.

Huang, P.-S.; He, X.; Gao, J.; Deng, L.; Acero, A.; and Heck, L. 2013. Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data. In *CIKM*, 2333–2338.

Jiao, X.; Yin, Y.; Shang, L.; Jiang, X.; Chen, X.; Li, L.; Wang, F.; and Liu, Q. 2020. TinyBERT: Distilling BERT for Natural Language Understanding. In *EMNLP Findings*, 4163–4174.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; and Soricut, R. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *ICLR*.

Lavie, T.; Sela, M.; Oppenheim, I.; Inbar, O.; and Meyer, J. 2010. User Attitudes towards News Content Personalization. *International Journal of Human-Computer Studies*, 68(8): 483–495.

Li, B.; Zhou, H.; He, J.; Wang, M.; Yang, Y.; and Li, L. 2020. On the Sentence Embeddings from Pre-trained Language Models. In *EMNLP*, 9119–9130.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lu, W.; Jiao, J.; and Zhang, R. 2020. TwinBERT: Distilling Knowledge to Twin-Structured Compressed BERT Models for Large-Scale Retrieval. In *CIKM*, 2645–2652.

Mirzadeh, S.-I.; Farajtabar, M.; Li, A.; Levine, N.; Matsukawa, A.; and Ghasemzadeh, H. 2020. Improved Knowledge Distillation via Teacher Assistant. In *AAAI*, 5191–5198.

Okura, S.; Tagami, Y.; Ono, S.; and Tajima, A. 2017. Embedding-based news recommendation for millions of users. In *KDD*, 1933–1942.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Sun, C.; Qiu, X.; Xu, Y.; and Huang, X. 2019a. How to Fine-Tune BERT for Text Classification? In *CCL*, 194–206.

Sun, S.; Cheng, Y.; Gan, Z.; and Liu, J. 2019b. Patient Knowledge Distillation for BERT Model Compression. In *EMNLP-IJCNLP*, 4323–4332.

Sun, Z.; Yu, H.; Song, X.; Liu, R.; Yang, Y.; and Zhou, D. 2020. MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices. In *ACL*, 2158–2170.

Tang, R.; Lu, Y.; Liu, L.; Mou, L.; Vechtomova, O.; and Lin, J. 2019a. Distilling Task-Specific Knowledge from BERT into Simple Neural Networks. *arXiv preprint arXiv:1903.12136*.

Tang, R.; Lu, Y.; Liu, L.; Mou, L.; Vechtomova, O.; and Lin, J. 2019b. Distilling Task-Specific Knowledge from BERT into Simple Neural Networks. *arXiv preprint arXiv:1903.12136*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS*, 5998–6008.

Wang, H.; Zhang, F.; Xie, X.; and Guo, M. 2018. DKN: Deep Knowledge-Aware Network for News Recommendation. In *WWW*, 1835–1844.

Wang, W.; Wei, F.; Dong, L.; Bao, H.; Yang, N.; and Zhou, M. 2020. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. In *NeurIPS*, 5776–5788.

Wu, C.; Wu, F.; An, M.; Huang, J.; Huang, Y.; and Xie, X. 2019a. Neural News Recommendation with Attentive Multi-View Learning. In *IJCAI*, 3863–3869.

Wu, C.; Wu, F.; An, M.; Huang, J.; Huang, Y.; and Xie, X. 2019b. NPA: neural news recommendation with personalized attention. In *KDD*, 2576–2584.

Wu, C.; Wu, F.; Ge, S.; Qi, T.; Huang, Y.; and Xie, X. 2019c. Neural News Recommendation with Multi-Head Self-Attention. In *EMNLP-IJCNLP*, 6389–6394.

Wu, C.; Wu, F.; and Huang, Y. 2021. One Teacher is Enough? Pre-trained Language Model Distillation from Multiple Teachers. In *Findings of ACL-IJCNLP*.

Wu, C.; Wu, F.; Qi, T.; and Huang, Y. 2021a. Empowering News Recommendation with Pre-Trained Language Models. In *SIGIR*, 1652–1656.

Wu, C.; Wu, F.; Yu, Y.; Qi, T.; Huang, Y.; and Liu, Q. 2021b. NewsBERT: Distilling Pre-trained Language Model for Intelligent News Application. In *Findings of EMNLP*.

Wu, F.; Qiao, Y.; Chen, J.-H.; Wu, C.; Qi, T.; Lian, J.; Liu, D.; Xie, X.; Gao, J.; Wu, W.; and Zhou, M. 2020. MIND: A Large-scale Dataset for News Recommendation. In *ACL*, 3597–3606.

Xiao, S.; Liu, Z.; Shao, Y.; Di, T.; and Xie, X. 2021. Training Microsoft News Recommenders with Pretrained Language Models in the Loop. *arXiv preprint arXiv:2102.09268*.

Xu, C.; Zhou, W.; Ge, T.; Wei, F.; and Zhou, M. 2020. BERT-of-Theseus: Compressing BERT by Progressive Module Replacing. In *EMNLP*, 7859–7869.

Zhang, Q.; Li, J.; Jia, Q.; Wang, C.; Zhu, J.; Wang, Z.; and He, X. 2021. UNBERT: User-News Matching BERT for News Recommendation. In *IJCAI*, 3356–3362.

Zhu, Q.; Zhou, X.; Song, Z.; Tan, J.; and Guo, L. 2019. Dan: Deep attention neural network for news recommendation. In *AAAI*, 5973–5980.

Appendix

Experiment Settings

In the domain-specific post-training experiments, 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. In the news recommendation experiments, 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 3.

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 K	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_1	1
Temperature T_2	1
β_1	1
β_2	0.1
Number of teacher models M	4
Initial value of ω	1

Table 3: Hyper-parameter settings

Additional Results on *MIND*

We also report the additional results on the *MIND* dataset, which are shown in Figs. 7-9. We observe similar phenomena to the results on the *Feeds* dataset.

Effectiveness of Each Loss Function

In this section, we conduct experiments to demonstrate the effectiveness of each part of the overall loss function in our two-stage knowledge distillation method, i.e. the distillation loss, the embedding loss, and the target loss. We compare the performance of the student models distilled with our Tiny-NewsRec approach and its variant with one part of the loss function in the second stage removed. The results on the *Feeds* and *MIND* datasets are shown in Fig.10. From the results, we have several findings. First, the distillation loss is the most critical part of the loss function as the performance declines significantly when 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 on certain training samples. The supervision from the ground-truth label is still necessary for the student model.

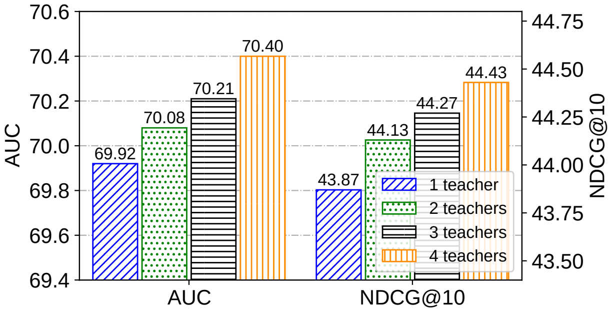


Figure 7: Impact of different number of teacher models.

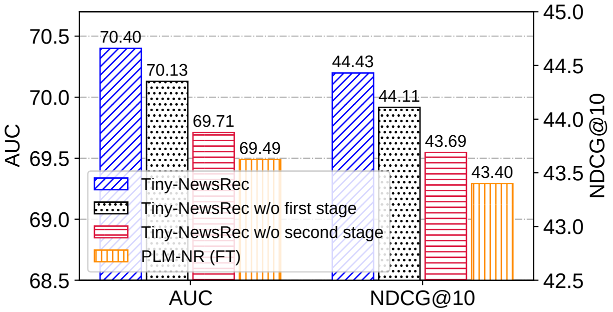


Figure 8: Effectiveness of each stage in our framework.

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 first-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.

Running Time

On the *News* dataset, the domain-specific post-training of the 12-layer teacher model and the first-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 finetuning of the 12-layer teacher model and the second-stage knowledge distillation of the 4-layer, 2-layer, and 1-layer

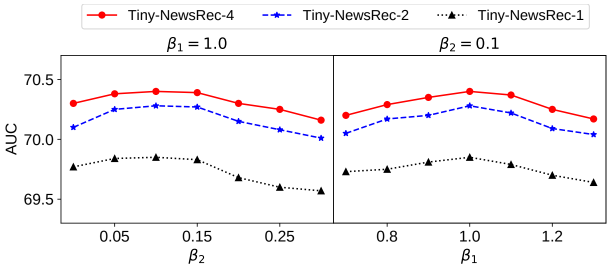
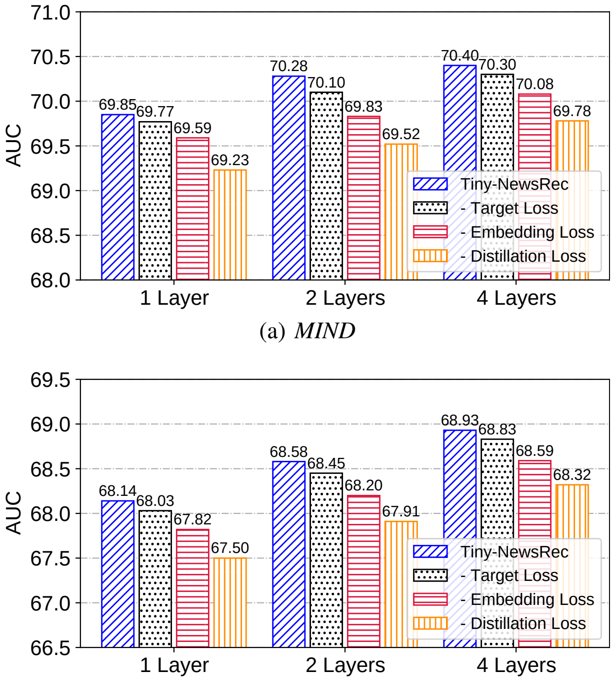


Figure 9: Influence of the hyper-parameter β_1 and β_2.



(a) *MIND*

(b) *Feeds*

Figure 10: Effectiveness of each loss function.

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.