Revisiting Language Models in Neural News Recommender Systems

Yuyue Zhao[1] [0000-0002-5298-0309], Jin Huang[1,2†] [0000-0001-9273-9037], David Vos[1] [0009-0003-8925-1585], and Maarten de Rijke[1] [0000-0002-1086-0202]

[1] University of Amsterdam
[2] Delft University of Technology
y.zhao3@uva.nl, j.huang2@uva.nl, d.j.a.vos@uva.nl, m.derijke@uva.nl

Abstract. Neural news recommender systems (RSs) have integrated language models (LMs) to encode news articles with rich textual information into representations, thereby improving the recommendation process. Most studies suggest that (i) news RSs achieve better performance with larger pre-trained language models (PLMs) than shallow language models (SLMs), and (ii) that large language models (LLMs) outperform PLMs. However, other studies indicate that PLMs sometimes lead to worse performance than SLMs. Thus, it remains unclear whether using larger LMs consistently improves the performance of news RSs. In this paper, we revisit, unify, and extend these comparisons of the effectiveness of LMs in news RSs using the real-world MIND dataset. We find that (i) larger LMs do not necessarily translate to better performance in news RSs, and (ii) they require stricter fine-tuning hyperparameter selection and greater computational resources to achieve optimal recommendation performance than smaller LMs. On the positive side, our experiments show that larger LMs lead to better recommendation performance for cold-start users: they alleviate dependency on extensive user interaction history and make recommendations more reliant on the news content.

Keywords: News recommendation · Language model · Fine-tuning.

1 Introduction

News recommender systems (RSs) help deliver relevant news articles to users. Unlike RSs in other domains, such as e-commerce and music, that primarily focus on modeling interactions between users and items, news RSs rely heavily on modeling text-based news articles with rich textual information [16]. Therefore, natural language processing techniques, particularly methods based on language models (LMs), are widely used to generate news representations in news RSs.

Among the early LM-based approaches to news representation are shallow language models (SLMs) such as GloVe [22], a model that generates word representations based on corpus co-occurrence statistics. In news RSs, GloVe embeddings are used to initialize word embeddings, which are later employed to model

† Corresponding author.

news articles and interactions [1, 27, 28]. Following progress in language modeling, pre-trained language models (PLMs) such as BERT [4] and RoBERTa [21] have also been integrated into news RSs to generate embeddings for news articles. Compared to SLMs, PLMs are typically larger, featuring complex architectures with more layers, thus contributing to a greater number of parameters. Large language models (LLMs) such as Llama [24] are also used to enhance news modeling in RSs because of their ability to capture context and to generalize [18, 20].

Most prior work in news RSs shows that news RSs using larger PLMs outperform those using SLMs [9, 15, 30–32, 35, 37, 38]. Work using LLMs has shown better recommendation performance than PLMs [18, 20]. But the findings are not consistent: some work reports that PLMs sometimes perform worse than SLMs in news RSs [7, 8]. Are larger models worth the additional computational resources? We examine (i) whether larger LMs truly improve news recommendation performance, and (ii) what size LMs (as news encoders) provides a reasonable trade-off between performance and resource consumption.

To answer these questions, we compare the impact of using eight LMs – across different LM families, i.e., GloVe (SLM), BERT and RoBERTa (PLMs), and Llama3.1-8B (LLM), as well as multiple sizes within the BERT family (tiny, mini, small, medium, and base) – on the performance of three well-known news recommendation models: NAML [27], NRMS [28], and LSTUR [1]. Consistent with widely adopted practices [3, 18, 20, 39], our experiments are based on the small version of the real-world MIND dataset [33].

We focus on the following research questions:

RQ1 Does using a larger LM in news RSs consistently lead to better recommendation accuracy?

RQ2 How does fine-tuning LMs affect the performance of LM-based news RSs?

LMs may enhance the performance of news RSs for cold-start users with limited or no user interaction history by analyzing the textual content of news articles and recommending relevant content based on extracted semantic information [23, 26]. Therefore, our third research question concerns the recommendation performance of different LMs for cold-start users:

RQ3 Do news RSs based on larger LMs provide better performance for cold-start users?

Larger LMs in news RSs do not always lead to improved performance of recommendations. The performance of LM-based news RSs depends heavily on whether the LMs are fine-tuned. E.g., without fine-tuning, NRMS using the SLM GloVe outperforms NRMS using PLMs BERT and RoBERTa, and even performs comparable to NRMS using LLM Llama. Moreover, while larger LMs require more comprehensive fine-tuning, such as searching for the optimal number of fine-tuned layers, they tend to achieve better performance for cold-start users. LM-enhanced news encoders alleviate the dependency on user interaction history, making recommendations more reliant on news content itself.

2 Related Work

Selection criteria for related work. We follow the guidelines in [13] to select relevant literature on LMs as news encoders for news recommendation. Sources are chosen from top venues and journals in the fields of artificial intelligence (AI) and information retrieval (IR). Papers are included if they (i) propose a definition of text modeling in the context of news recommendation, (ii) introduce approaches to improve news recommendation performance, or (iii) present experimental results comparing the performance of different-sized LMs as news encoders using the same benchmark. Papers are excluded if (i) their approaches are not tested on an English news recommendation dataset or (ii) they fall outside the date range of October 2014 (the release of GloVe) to October 2024. We identified over 200 studies, 24 of which are highly relevant to our work. Below, we introduce these studies to provide context for our research.

LMs in news RSs. News content modeling is a crucial component of news RSs, as news articles contain rich textual information that can be effectively encoded using LMs [29]. Following the categorization criteria in [19, 34], methods using LMs in news RSs can be grouped based on the role of the LM: (i) LMs as news recommenders, which generate candidate news items [17, 18], (ii) LMs as news encoders, which encode news content to support news RSs [7, 9, 15, 30–32, 35, 37, 38], and (iii) LMs as news enhancers, which generate additional textual features that assist news RSs [20, 36]. In this study, we focus on the largest group, where LMs are used as news encoders to explore the impact of different LMs in news RSs in relation to their effectiveness and efficiency.

LMs as news encoders. Early LM-based approaches to news RSs learn representations on SLMs, such as GloVe. NAML [27] uses GloVe embeddings to initialize word representations and employs a word-level and view-level attention mechanism, along with convolutional neural networks, to capture important words for news representation. NRMS [28] uses GloVe embeddings for initialization and adopts multi-head self-attention to learn news representation. Prior work has argued that such shallow models may not be sufficient to capture the semantic information in news articles, and has explored PLMs based on the transformer architecture [25], such as BERT, for news modeling. E.g., PLM-NR [30] uses PLMs to enhance news representation and observes improvements over SLMs as a news encoder model. MINER [15] employs a pre-trained BERT as the news encoder and uses a poly-attention mechanism to extract multiple aspect interest vectors for users. More recently, LLMs have been explored for news modeling. ONCE [20] uses both open- and closed-source LLMs to enrich training data and enhance content representation. Yada and Yamana [36] improve news recommendations by using LLMs to generate category descriptions. PGNR [18] employs LLMs to frame news recommendation as a text-to-text generation task, performing recommendation through generation.

According to the studies listed above, transitioning from SLMs to PLMs and then to LLMs results in clear improvements in news recommendation performance. However, some studies report different findings. NewsRecLib [7], a widely used news recommendation benchmark, reports that NAML and LSTUR,

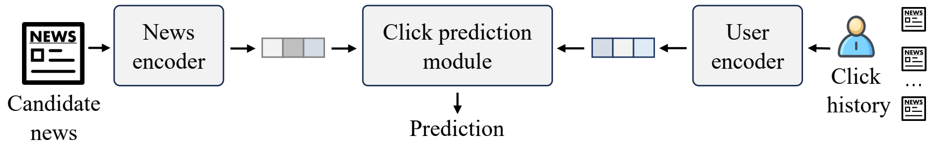


Fig. 1: The typical structure of neural news recommendation methods.

which originally used GloVe, performs worse when GloVe is replaced by the PLM BERT. Additionally, xMIND [8], a publicly available multilingual benchmark for news recommendation, indicates that NAML using PLM-based embeddings performs worse than the version using randomly-initialized embeddings. These contradictory results highlight the need for further investigation into the effectiveness of LMs in news recommendation, which motivates this study.

3 Reproducibility Methodology

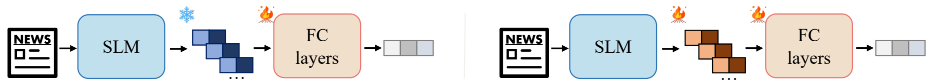
3.1 Problem formulation

Let $\mathcal{V}$ represent the set of news articles, where each news article $v \in \mathcal{V}$ consists of its textual feature $f_t(v)$ (e.g., title or abstract) and other features $f_d(v)$ (e.g., news categories or subcategories). Let $\mathcal{U}$ represent the set of users. Each user $u \in \mathcal{U}$ has a click history $H_u = \{v_1^h, v_2^h, \ldots, v_n^h\}$ in chronological order, denoting the sequence of n news articles previously clicked by the user. Given a candidate news article $v_c \in \mathcal{V}$, the goal of a news recommendation method is to predict the probability $\hat{y}_{u,v_c}$ that user u will click on v_c.

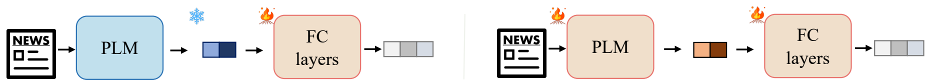
A typical (neural) news recommendation method has three components: a news encoder, a user encoder, and a click prediction module, as shown in Fig. 1. The news encoder, primarily based on LMs, is responsible for encoding the textual features $f_t(v)$ and/or other features $f_d(v)$ associated with news article v, ultimately producing the news representation $\boldsymbol{q}_v$. We focus solely on the textual features $f_t(v)$ to compare the ability of different LMs in news modeling within news RSs. The user encoder generates the user preference representation $\boldsymbol{p}_u$ based on the user's click history H_u, summarizing the representations of news articles they have browsed. Using these representations, the click prediction module estimates the click probability $\hat{y}_{u,v_c}$ for candidate news article v_c.

3.2 News recommendation methods

Following [7, 9, 15, 30, 35, 37], we select NAML [27], NRMS [28], and LSTUR [1] as (neural) news recommendation systems; all involve attention mechanisms for news recommendation. In terms of the news encoder, all three use attention for news modeling; NAML incorporates different types of news information, such as titles, bodies, categories, and subcategories, while NRMS focuses solely on learning news representations from titles. LSTUR models news representations based on titles and topic categories. NAML and NRMS employ attention mechanisms to learn user representations, whereas LSTUR uses a GRU network.



(a) SLMs in news encoders.



(b) PLMs in news encoders.

Fig. 2: SLMs and PLMs as building blocks of news encoders. Each LM can be used either in its non-fine-tuned form, shown in the left plots, or in its fine-tuned form, shown in the right plots. The parameters/embeddings in the blue "ice" section are fixed, while those in the red "flame" section are fine-tuned.

As depicted in Fig. 1, the news encoder takes news features as input and then yields the news representation $\boldsymbol{q}_v$. The user encoder learns the user representation $\boldsymbol{p}_u$ based on the user's click history H_u, and the click score $\hat{y}$ is computed following the click prediction module $\mathcal{F}^{\mathrm{RS}}(\cdot)$:

$$\hat{y}_{u,v} = \mathcal{F}^{\mathrm{RS}}([\boldsymbol{p}_u, \boldsymbol{q}_v]). \tag{1}$$

For model training, the loss function minimizes the negative log-likelihood of all positive news articles in the ground truth:

$$\mathcal{L} = -\sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{V}^+} \log \frac{\exp(\hat{y}_{u,i})}{\exp(\hat{y}_{u,i}) + \sum_{j \in \mathcal{V}^-} \exp(\hat{y}_{u,j})}, \tag{2}$$

where $\mathcal{V}^+$ is the set of positive new articles for user u in the training dataset, and $\mathcal{V}^-$ is the sampled negative news set corresponding to user u and the i-th positive news. This optimization encourages the model to differentiate between clicked and non-clicked news articles.

3.3 Language models as news encoders

To investigate the effect of different LMs as news encoders on the performance of news RSs, we compare three types of LMs based on model size: SLMs, PLMs, and LLMs. Each LM, when used as a news encoder, can either be used in its *non-fine-tuned* form, relying on its pre-trained knowledge, or it can be *fine-tuned* with additional training on news-specific data to improve the performance on the recommendation task.

SLMs as news encoders. Given a news article $v = [v_1, v_2, \ldots, v_n]$, where v_i represents the i-th word in article v, SLMs generate static, non-contextualized word embeddings $\boldsymbol{e}_{v_i}$ by aggregating global word co-occurrence statistics, with each word having its embedding regardless of context: $\boldsymbol{e}_{v_i} = \mathcal{M}^{\mathrm{SLM}}(v_i; \theta^{\mathrm{SLM}})$. The news representation $\boldsymbol{q}_v$ is obtained by concatenating the word embeddings of the news content and then applying a fully-connected layer (FC): $\boldsymbol{q}_v = \mathrm{FC}([\boldsymbol{e}_{v_1} \| \boldsymbol{e}_{v_2} \| \cdots \| \boldsymbol{e}_{v_n}]; \theta^{\mathrm{FC}})$, where $\|$ denotes the concatenation operator.[3]

[3] All FC layers in the paper share a similar architecture, differing primarily in the size of the first layer, which varies depending on the input size to enable the news RS to process varying input dimensions.

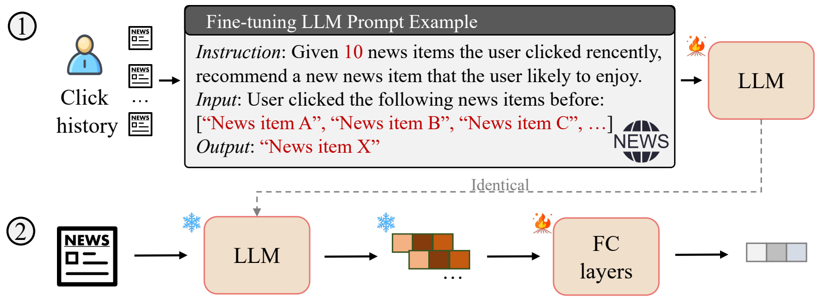


Fig. 3: Fine-tuning LLMs as news encoders. In step 1, the LLMs are fine-tuned on news data presented in a natural language format. In step 2, the fine-tuned LLMs generate news embeddings, which are used for the recommendation task.

Non-fine-tuned mode. As illustrated in the left part of Fig. 2a, in the non-fine-tuned mode, SLMs map each word in a news article to its corresponding embedding $\boldsymbol{e}_{v_i}$, which are then concatenated for further processing. The parameters of the fully connected layer θ^{FC} are tuned according to Eq. 2.

Fine-tuned mode. As illustrated in the right part of Fig. 2a, in the fine-tuned mode, the word embeddings $\boldsymbol{e}_{v_i}$ and fully-connected layer parameters θ^{FC} are updated based on the recommendation signal (see Eq. 2).

PLMs as news encoders. For a news article v, PLMs first tokenize the news text into tokens $\mathcal{T}^{\mathrm{PLM}}(v) = [v^t_{[\mathrm{CLS}]}, v^t_1, \ldots, v^t_m, v^t_{[\mathrm{SEP}]}]$. The PLMs then generate contextualized token embeddings by passing the token sequence through transformer encoder layers: $[\boldsymbol{e}_{v^t_{[\mathrm{CLS}]}}, \boldsymbol{e}_{v^t_1}, \ldots, \boldsymbol{e}_{v^t_m}, \boldsymbol{e}_{v^t_{[\mathrm{SEP}]}}] = \mathcal{M}^{\mathrm{PLM}}([v^t_{[\mathrm{CLS}]}, v^t_1, \ldots, v^t_m, v^t_{[\mathrm{SEP}]}]; \theta^{\mathrm{PLM}})$. Following [15, 30, 35], we use the embedding of the `[CLS]` token, which appears at the start of every input sequence, and apply a fully connected layer to represent the entire news article: $\boldsymbol{q}_v = \mathrm{FC}(\boldsymbol{e}_{v^t_{[\mathrm{CLS}]}}; \theta^{\mathrm{FC}})$.

Non-fine-tuned mode. As shown in the left part of Fig. 2b, a PLM models the news content and outputs the news embedding. Similar to SLMs, the parameters of the fully-connected layer θ^{FC} are trained using the loss in Eq. 2.

Fine-tuned mode. As illustrated in the right part of Fig. 2b, both the PLM parameters θ^{PLM} and the fully-connected layer parameters θ^{FC} are updated during the recommendation process.

LLMs as news encoders. For LLMs, given a news article v, we follow the approach in [10], using fill-in-the-blank prompts, *i.e.*, "*This news: [v] means in one word:*" to create a prompted version of the news, denoted as v'. We tokenize it with $\mathcal{T}^{\mathrm{LLM}}(v') = [{v'}^t_1, {v'}^t_2, \ldots, {v'}^t_L]$. Then the LLM generates token embeddings: $[\boldsymbol{e}_{{v'}^t_1}, \boldsymbol{e}_{{v'}^t_2}, \ldots, \boldsymbol{e}_{{v'}^t_L}] = \mathcal{M}^{\mathrm{LLM}}([{v'}^t_1, {v'}^t_2, \ldots, {v'}^t_L]; \theta^{\mathrm{LLM}})$. We use the embeddings of the last l tokens[4] and apply a fully-connected layer, representing the news article: $\boldsymbol{q}_v = \mathrm{FC}([\boldsymbol{e}_{{v'}^t_{L-l}}, \ldots, \boldsymbol{e}_{{v'}^t_L}]; \theta^{\mathrm{FC}})$.

Non-fine-tuned mode. Similar to SLMs and PLMs, the parameters of the fully-connected layer θ^{FC} are trained in this mode.

[4] l is set to 10 in practice due to computational efficiency.

Table 1: Statistics of the MIND dataset.

#users	#news	#words in title	#words in abs	#pos clicks	#neg clicks
94,057	65,238	11.79	38.17	347,727	8,236,715

Fine-tuned mode. Due to a large number of parameters and high computational costs, as illustrated in Fig. 3, we adopt a two-step process inspired by [2]. First, news recommendation data is transformed into a natural language prompt format, and the LLM parameters θ^{LLM} are updated using cross-entropy loss to let the LLM learn news recommendation-specific information. Second, the LLM, which is fine-tuned in the first step and fixed in the second step, outputs the news embedding for the recommendation process in the same way as in the non-fine-tuned mode, with the fully-connected layer parameters θ^{FC} being updated according to the recommendation objective.

4 Experimental Setup

Below, we detail the dataset and implementation; resources to reproduce our results are available at https://github.com/Go0day/LM4newsRec.

4.1 The MIND dataset

Following [1, 3, 18, 20, 27, 28, 39], we conduct experiments using the MIND [33] dataset, a public news recommendation dataset collected from the Microsoft News website. Table 1 provides descriptive statistics for the dataset. We use the small version of the original MIND dataset, which is widely adopted in academic research and consists of randomly sampled users and their behavior logs. Impressions from November 9 to 14, 2019 are used for training, and those from November 15, 2019 are used for testing [18, 33].

4.2 Implementation details

We use four representative LMs: *GloVe.840B.300d*[5] (referred to as GloVe), *bert-base-uncased*[6] (BERT base version, 110M parameters), *roberta-base*[7] (RoBERTa, 125M), and *Llama 3.1-8B*[8] (Llama for short). These models are selected to cover different families of LMs. To examine the impact of varying model sizes within the same family, we further explore the BERT family by comparing different versions: BERT_{tiny} (4.4M parameters), BERT_{mini} (11.3M), BERT_{small} (29.1M), and BERT_{medium} (41.7M).[9]

Among the selected models, *GloVe.840B.300d* is an SLM, *Llama 3.1-8B* an LLM, and the rest are PLMs. For the PLMs, we fine-tune varying numbers of layers (from none to all) and select the optimal configuration based on recommendation performance. For Llama, we apply LoRA [6] for fine-tuning in step 1,

[5] https://nlp.stanford.edu/projects/glove [6] https://huggingface.co/google-bert
[7] https://huggingface.co/FacebookAI [8] https://huggingface.co/meta-llama
[9] https://huggingface.co/prajjwal1

then pre-compute and store news embeddings in advance for recommendation in step 2 (see Fig. 3). Specifically, NAML [27], NRMS [28], and LSTUR [1] were originally equipped with GloVe, while PLM-NR [30] was originally equipped with the BERT base version. We re-implement these foundational publications, standardize them within a unified news recommendation setting (including consistent datasets and model structures), and extend their evaluation by incorporating different LMs. For all recommendation methods, the maximum length of news titles is set to 20 tokens, and for news abstracts, it is set to 50 tokens; we search the size of the negative clicked news set $|\mathcal{V}^-|$ in Eq. 2 from {1, 2, 3, 4}, the dropout ratio from {0.1, 0.2, 0.3, 0.4, 0.5}, and the learning rate from {0.0001, 0.00001}. We use AUC, MRR, nDCG@5 (N@5), and nDCG@10 (N@10) as our evaluation metrics.

5 Results

5.1 RQ1: Impact of LMs on news recommendation accuracy

To answer RQ1, we train news RS methods with different sizes of LMs, as detailed in Section 3.3. The results are reported in Table 2. We observe:

(1) GloVe generally yields the lowest performance across different LM families, which is expected given its shallow structure. However, it surpasses BERT variants (BERT$_{tiny}$, BERT$_{mini}$, and BERT$_{small}$), showing that larger LMs do not inherently guarantee superior performance as news encoders.
(2) Comparing BERT and RoBERTa, BERT outperforms RoBERTa in most cases, except on LSTUR. This suggests that BERT may offer more effective news encoding, despite RoBERTa's higher parameter count.
(3) The performance of Llama does not significantly exceed that of other LMs, despite its considerably larger parameter count. Thus, an increase in parameters alone does not necessarily translate to better performance.
(4) Within the BERT family, larger models generally achieve better performance than smaller variants. There is one exception: BERT$_{small}$ does not consistently outperform BERT$_{mini}$, even with a higher parameter count.

Our findings for RQ1 indicate that, across different LM families, larger LMs do *not* consistently improve news recommendation performance. Within the BERT family, models with more parameters generally perform better; however, this trend is not absolute, as seen in the performance of BERT$_{mini}$ versus BERT$_{small}$.

5.2 RQ2: Impact of fine-tuning LMs on performance and efficiency

To investigate the role of fine-tuning, we compare the news recommendation accuracy of non-fine-tuned vs. fine-tuned models and evaluate the computational efficiency of different LMs. This section highlights the trade-off between improved performance and computational feasibility.

Effectiveness. Table 3 and Fig. 4 show that fine-tuning LMs generally improves news recommendation performance, highlighting the effectiveness of fine-tuning.

Table 2: Performance comparison of different LMs as news encoders deployed across three news recommendation methods on the MIND dataset. "BERT" in the left section denotes BERT base version. Results are averaged over three runs and reported as percentages (%). Bold font indicates the winner in that column.

		Performance					Performance			
Model	LM	AUC	MRR	N@5	N@10	LM	AUC	MRR	N@5	N@10
NAML	GloVe	66.29	31.61	34.93	41.22	BERT$_{tiny}$	64.83	30.71	33.89	40.24
	BERT	<u>67.30</u>	<u>32.62</u>	<u>36.04</u>	<u>42.19</u>	BERT$_{mini}$	<u>65.99</u>	31.58	34.76	41.02
	RoBERTa	66.73	32.10	35.52	41.64	BERT$_{small}$	65.91	<u>31.70</u>	<u>34.89</u>	<u>41.24</u>
	Llama	**68.39**	**33.20**	**36.88**	**43.06**	BERT$_{medium}$	**67.03**	**32.50**	**35.97**	**42.06**
NRMS	GloVe	66.62	31.34	34.83	41.04	BERT$_{tiny}$	64.30	29.10	31.76	38.56
	BERT	**68.05**	**31.80**	**35.30**	**41.72**	BERT$_{mini}$	<u>65.70</u>	<u>30.34</u>	<u>33.32</u>	39.82
	RoBERTa	<u>67.22</u>	<u>31.59</u>	<u>34.94</u>	<u>41.35</u>	BERT$_{small}$	65.60	30.17	33.19	<u>39.83</u>
	Llama	66.64	31.66	35.05	41.33	BERT$_{medium}$	**66.83**	**31.20**	**34.49**	**40.95**
LSTUR	GloVe	60.43	26.26	28.62	35.11	BERT$_{tiny}$	58.48	24.47	26.41	33.09
	BERT	<u>60.92</u>	26.74	29.14	<u>35.55</u>	BERT$_{mini}$	58.80	<u>24.81</u>	<u>27.03</u>	<u>33.62</u>
	RoBERTa	**61.25**	**27.15**	**29.60**	**35.84**	BERT$_{small}$	<u>59.14</u>	24.65	26.66	33.55
	Llama	60.88	<u>26.83</u>	<u>29.21</u>	35.54	BERT$_{medium}$	**59.66**	**25.39**	**27.93**	**34.38**

However, for Llama, the performance benefits of fine-tuning decline in NAML. A plausible reason is that NAML uses both title and abstract text, which may introduce redundancy and noise during Llama's two-step fine-tuning process.

Within the BERT family (see Fig. 4), we observe that all models benefit from fine-tuning. Notably, in non-fine-tuned settings, most BERT models do not outperform GloVe. This may be due to GloVe's pre-training on the Common Crawl web data [22], likely making it more suited to news content than BERT models trained on BookCorpus and Wikipedia [4]. The effectiveness of GloVe in representing news content could also explain why fine-tuning leads to a slight performance drop when used in NRMS (see Table 3).

Additionally, when analyzing BERT (base) by fine-tuning different numbers of layers as shown in Fig. 5, we observe that the optimal number of layers varies significantly across different RS methods. Interestingly, in some cases, fine-tuned BERT does not outperform non-fine-tuned GloVe, further underscoring GloVe's strength in capturing news representations.

Efficiency. Fig. 6 provides parameter statistics within the NAML framework, with similar trends in NRMS and LSTUR. "Total parameters" represents all parameters involved in the recommendation process, while "trainable parameters" includes only those updated during training (see Section 3.3). Generally, as LM size increases, both total and trainable parameters grow. However, for Llama, the two-step fine-tuning strategy and the use of pre-computed news embeddings significantly reduce its trainable parameters. And for GloVe, the concatenation of word embeddings results in a higher number of trainable parameters than BERT$_{tiny}$ and is comparable to BERT$_{mini}$.

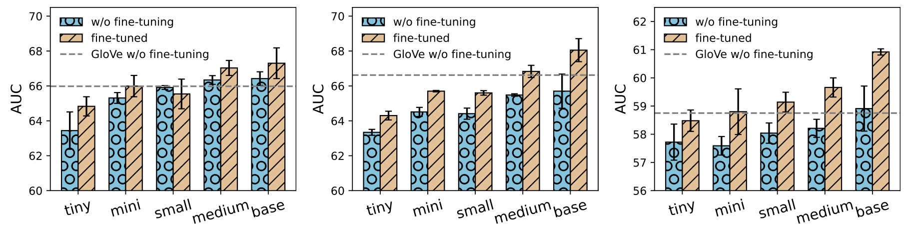
Table 3: Performance comparison between fine-tuned and non-fine-tuned settings. “Change” denotes AUC gain of fine-tuned LMs over non-fine-tuned ones.

Model	LM	Fine-tuned?	AUC	MRR	nDCG@5	nDCG@10	Change
NAML	GloVe	Y	66.29	31.61	34.93	41.22	+ 0.46%
		N	65.98	31.67	35.15	41.10	
	BERT	Y	67.30	32.62	36.04	42.19	+ 1.32%
		N	66.42	31.53	34.71	41.06	
	RoBERTa	Y	66.73	32.10	35.52	41.64	+ 5.07%
		N	63.51	29.25	32.29	38.70	
	Llama	Y	67.90	32.99	36.61	42.72	− 0.72%
		N	68.39	33.20	36.88	43.06	
NRMS	GloVe	Y	65.96	30.86	34.09	40.54	− 0.99%
		N	66.62	31.34	34.83	41.04	
	BERT	Y	68.05	31.80	35.30	41.72	+ 3.58%
		N	65.70	30.56	33.33	40.13	
	RoBERTa	Y	67.22	31.59	34.94	41.35	+ 9.11%
		N	61.61	26.44	29.02	35.77	
	Llama	Y	66.64	31.66	35.05	41.33	+ 0.11%
		N	66.56	31.71	35.13	41.30	
LSTUR	GloVe	Y	60.43	26.26	28.62	35.11	+ 2.86%
		N	58.75	25.66	27.89	34.20	
	BERT	Y	60.92	26.74	29.14	35.55	+ 3.41%
		N	58.91	25.17	27.19	33.91	
	RoBERTa	Y	61.25	27.15	29.60	35.84	+ 6.09%
		N	57.74	24.45	26.66	32.97	
	Llama	Y	60.88	26.83	29.21	35.54	+ 2.35%
		N	59.48	26.00	28.36	34.62	

Overall, these findings underscore different trade-offs when selecting a LM as news encoder (answering RQ2): GloVe is an efficient option for cases with limited fine-tuning resources, offering effective performance with minimal computational demands. For static datasets without frequent news updates, precomputing Llama embeddings and using them for inference offers a high-performance alternative. Finally, when both computational resources and performance are priorities, fine-tuning the BERT base version for the news recommendation task provides a balanced, high-performing solution.

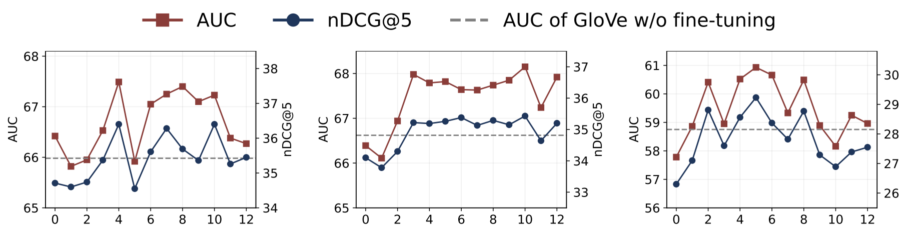
5.3 RQ3: Impact of LMs on cold-start user performance

Given the inconsistent results regarding performance gains with larger LMs as news encoders (see Section 5.1), we further investigate whether larger LMs benefit specific user groups in news RSs. To examine this, we test three representative LMs: GloVe, BERT (base), and Llama. The users are sorted by click history



(a) NAML. (b) NRMS. (c) LSTUR.

Fig. 4: Effect of fine-tuning versus no fine-tuning in the BERT family.



(a) NAML. (b) NRMS. (c) LSTUR.

Fig. 5: Effect of varying the number of fine-tuned layers in BERT.

length and categorized into five engagement levels: Group 1 (0–20%), Group 2 (20%–40%), Group 3 (40%–60%), Group 4 (60%–80%), and Group 5 (80%–100%), representing different levels of engagement based on the distribution of click history length. The average click lengths for each group, *i.e.,* the number of news articles previously clicked by users in each group, are 4.01, 9.27, 16.57, 29.60, and 48.58, respectively. In Fig. 7, the bars show the AUC scores for various LMs as news encoders across user groups, while the line plot illustrates each LM's relative change over GloVe.

We find that Llama provides the greatest improvement in Group 1, which includes the "coldest" users with the smallest amount of click history. This improvement may be due to the limited interaction data available for these users, where larger LMs can leverage richer text-based representations to alleviate sparse click signals. As users' click history expands (*e.g.,* Group 5), the relative benefit of larger LMs diminishes, indicating that user engagement itself provides a strong signal for modeling. Interestingly, in the LSTUR model (see Fig. 7c), the relative improvement from larger LMs decreases more sharply and even turns negative in Group 5. This may be attributed to LSTUR's use of GRU for modeling user preferences, which, unlike the attention mechanisms used in NAML and NRMS, is less effective at capturing evolving user interests [11]. As news representations become more comprehensive with larger LMs, the limitations in modeling dynamic user preferences likely contribute to the observed performance declines.

In response to RQ3, these findings suggest that larger LMs enhance performance for cold-start users. Their effectiveness decreases as click history increases, especially in news RSs with limited user modeling capabilities.

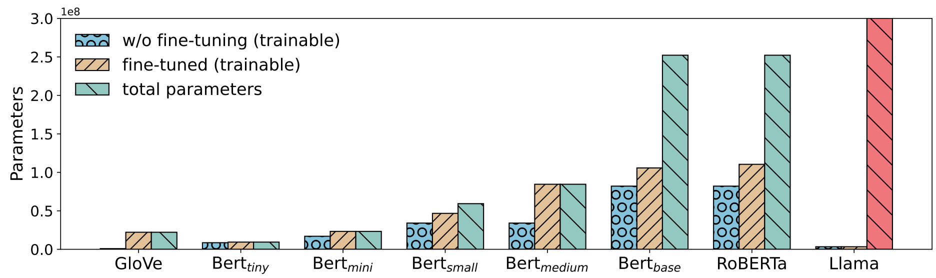
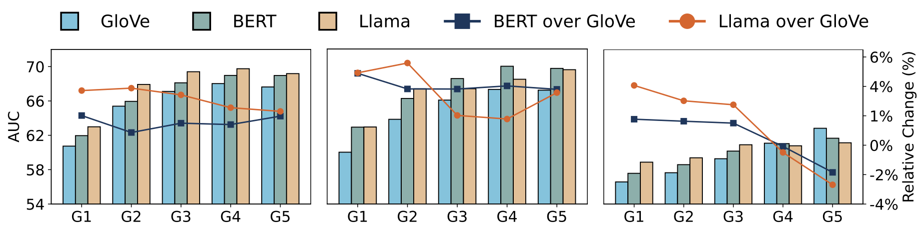


Fig. 6: Comparison of trainable and total parameters for different LMs in fine-tuned and non-fine-tuned modes within the NAML framework. Llama's total parameter (over 8 billion) is highlighted in red as it significantly exceeds the scale of other models and cannot be visually included within the same figure.



(a) NAML. (b) NRMS. (c) LSTUR.

Fig. 7: Effect of LMs across user groups with varying click history lengths. User groups '1' through '5' represent progressively longer click histories. The 'Relative Change' indicates each LM's performance improvement ratio compared to GloVe.

6 Limitations and Broader Impact

Our study has several limitations. First, we only use the MIND-small dataset for news recommendation due to resource constraints. Additionally, we limit our analysis to English news datasets to focus on evaluating model size effectiveness, leaving out datasets in other languages, such as EB-NeRD [14], Adressa [5], and Plista [12]. Investigating the impact of LMs in non-English news recommendation would be an interesting and valuable direction. Second, we examine LMs with a maximum of 8 billion parameters (Llama) as news encoders, as evaluating larger models (e.g., 13 billion, 70 billion, etc.) exceeded our resource capacity. We expect that larger models might offer further gains, particularly for cold-start users. Third, our study explores three news recommendation methods: NAML, NRMS, and LSTUR, which are commonly used as benchmarks [see, e.g., 29, 30, 37]. In news recommendation, a significant proportion of news articles that are awaiting recommendations do not appear in the logged data. Specifically, in the MIND dataset we used, approximately 32.9% of the news articles in the test set never appear in the training set. This makes ID-based collaborative filtering methods, such as matrix factorization and graph-based approaches, unsuitable for our setting. Therefore, we focus on these three representative content-based news recommendation methods and leave the exploration of other techniques for future work.

Beyond limitations, our study has broader impacts. It provides a reference point for both academia and industry regarding the role of LMs in news recommendation, showing that larger models do not always translate to better performance. Our findings demonstrate that deploying LMs can help address the cold-start problem for new users, enhancing recommendation reliability for underrepresented groups. We believe our work has the potential to contribute to advancing socially responsible and reliable news recommendation systems.

7 Conclusion

In this work, we have revisited the role of language models (LMs) as news encoders within neural news RSs on the MIND dataset. We have investigated the effects of varying LM sizes, assessing the impact of fine-tuning on recommendation performance and analyzing model performance across different user groups.

Our main finding is that larger LMs as news encoders do not consistently yield better recommendation results, contrasting with previous studies [30, 37]. Additionally, we observe that larger LMs require more precise fine-tuning and greater computational resources, prompting a trade-off consideration based on performance needs and resource availability.

Notably, we identify an interesting tendency: larger LMs show more significant improvements in recommendations for cold-start users, suggesting potential benefits in modeling user interests with limited click history. A promising future direction is to investigate the stability of LM-based RSs as the news RS domain evolves. Additionally, exploring the design of larger LMs to better meet the dynamic needs of diverse user groups would be valuable.

Acknowledgments

This research was (partially) supported by the Dutch Research Council (NWO), under project numbers 024.004.022, NWA.1389.20.183, and KICH3.LTP.20.006, and the European Union's Horizon Europe program under grant agreement No 101070212.

All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

Bibliography

[1] An, M., Wu, F., Wu, C., Zhang, K., Liu, Z., Xie, X.: Neural news recommendation with long-and short-term user representations. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 336–345 (2019)

[2] Bao, K., Zhang, J., Wang, W., Zhang, Y., Yang, Z., Luo, Y., Chen, C., Feng, F., Tian, Q.: A bi-step grounding paradigm for large language models in recommendation systems. arXiv preprint arXiv:2308.08434 (2023)

[3] Bi, Q., Li, J., Shang, L., Jiang, X., Liu, Q., Yang, H.: Mtrec: Multi-task learning over bert for news recommendation. In: Findings of the Association for Computational Linguistics: ACL 2022, pp. 2663–2669 (2022)

[4] Devlin, J.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

[5] Gulla, J.A., Zhang, L., Liu, P., Özgöbek, Ö., Su, X.: The Adressa dataset for news recommendation. In: Proceedings of the International Conference on Web Intelligence, pp. 1042–1048 (2017)

[6] Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)

[7] Iana, A., Glavaš, G., Paulheim, H.: NewsRecLib: A pytorch-lightning library for neural news recommendation. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 296–310 (2023)

[8] Iana, A., Glavaš, G., Paulheim, H.: Mind your language: A multilingual dataset for cross-lingual news recommendation. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 553–563 (2024)

[9] Jiang, J.: Tadi: Topic-aware attention and powerful dual-encoder interaction for recall in news recommendation. In: Findings of the Association for Computational Linguistics: EMNLP 2023, pp. 15647–15658 (2023)

[10] Jiang, T., Huang, S., Luan, Z., Wang, D., Zhuang, F.: Scaling sentence embeddings with large language models. arXiv preprint arXiv:2307.16645 (2023)

[11] Kang, W.C., McAuley, J.: Self-attentive sequential recommendation. In: 2018 IEEE international conference on data mining (ICDM), pp. 197–206, IEEE (2018)

[12] Kille, B., Hopfgartner, F., Brodt, T., Heintz, T.: The plista dataset. In: Proceedings of the 2013 International News Recommender Systems Workshop and Challenge, pp. 16–23 (2013)

[13] Kitchenham, B., et al.: Guidelines for performing systematic literature reviews in software engineering. Tech. rep., Keele University (2007), version 2.3

[14] Kruse, J., Lindskow, K., Kalloori, S., Polignano, M., Pomo, C., Srivastava, A., Uppal, A., Andersen, M.R., Frellsen, J.: EB-NeRD: A large-scale dataset for news recommendation. arXiv preprint arXiv:2410.03432 (2024)
[15] Li, J., Zhu, J., Bi, Q., Cai, G., Shang, L., Dong, Z., Jiang, X., Liu, Q.: Miner: Multi-interest matching network for news recommendation. In: Findings of the Association for Computational Linguistics: ACL 2022, pp. 343–352 (2022)
[16] Li, M., Wang, L.: A survey on personalized news recommendation technology. IEEE Access **7**, 145861–145879 (2019)
[17] Li, X., Zhang, Y., Malthouse, E.C.: Exploring fine-tuning chatgpt for news recommendation. arXiv preprint arXiv:2311.05850 (2023)
[18] Li, X., Zhang, Y., Malthouse, E.C.: Prompt-based generative news recommendation (pgnr): Accuracy and controllability. In: European Conference on Information Retrieval, pp. 66–79, Springer (2024)
[19] Lin, J., Dai, X., Xi, Y., Liu, W., Chen, B., Zhang, H., Liu, Y., Wu, C., Li, X., Zhu, C., et al.: How can recommender systems benefit from large language models: A survey. arXiv preprint arXiv:2306.05817 (2023)
[20] Liu, Q., Chen, N., Sakai, T., Wu, X.M.: ONCE: Boosting content-based recommendation with both open-and closed-source large language models. In: Proceedings of the 17th ACM International Conference on Web Search and Data Mining, pp. 452–461 (2024)
[21] Liu, Y.: RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
[22] Pennington, J., Socher, R., Manning, C.D.: GloVe: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
[23] Sanner, S., Balog, K., Radlinski, F., Wedin, B., Dixon, L.: Large language models are competitive near cold-start recommenders for language-and item-based preferences. In: Proceedings of the 17th ACM Conference on Recommender Systems, pp. 890–896 (2023)
[24] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
[25] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
[26] Wang, J., Lu, H., Caverlee, J., Chi, E.H., Chen, M.: Large language models as data augmenters for cold-start item recommendation. In: Companion Proceedings of the ACM on Web Conference 2024, pp. 726–729 (2024)
[27] Wu, C., Wu, F., An, M., Huang, J., Huang, Y., Xie, X.: Neural news recommendation with attentive multi-view learning. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, pp. 3863–3869 (2019)
[28] Wu, C., Wu, F., Ge, S., Qi, T., Huang, Y., Xie, X.: Neural news recommendation with multi-head self-attention. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th

International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 6389–6394 (2019)
[29] Wu, C., Wu, F., Huang, Y., Xie, X.: Personalized news recommendation: Methods and challenges. ACM Transactions on Information Systems **41**(1), 1–50 (2023)
[30] Wu, C., Wu, F., Qi, T., Huang, Y.: Empowering news recommendation with pre-trained language models. In: Proceedings of the 44th international ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1652–1656 (2021)
[31] Wu, C., Wu, F., Qi, T., Huang, Y.: Are big recommendation models fair to cold users? arXiv preprint arXiv:2202.13607 (2022)
[32] Wu, C., Wu, F., Yu, Y., Qi, T., Huang, Y., Liu, Q.: NewsBERT: Distilling pre-trained language model for intelligent news application. arXiv preprint arXiv:2102.04887 (2021)
[33] Wu, F., Qiao, Y., Chen, J.H., Wu, C., Qi, T., Lian, J., Liu, D., Xie, X., Gao, J., Wu, W., et al.: MIND: A large-scale dataset for news recommendation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 3597–3606 (2020)
[34] Wu, L., Zheng, Z., Qiu, Z., Wang, H., Gu, H., Shen, T., Qin, C., Zhu, C., Zhu, H., Liu, Q., et al.: A survey on large language models for recommendation. World Wide Web **27**(5), 60 (2024)
[35] Xiao, S., Liu, Z., Shao, Y., Di, T., Middha, B., Wu, F., Xie, X.: Training large-scale news recommenders with pretrained language models in the loop. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 4215–4225 (2022)
[36] Yada, Y., Yamana, H.: News recommendation with category description by a large language model. arXiv preprint arXiv:2405.13007 (2024)
[37] Yi, J., Wu, F., Wu, C., Liu, R., Sun, G., Xie, X.: Efficient-FedRec: Efficient federated learning framework for privacy-preserving news recommendation. arXiv preprint arXiv:2109.05446 (2021)
[38] Zhang, Q., Li, J., Jia, Q., Wang, C., Zhu, J., Wang, Z., He, X.: UNBERT: User-news matching BERT for news recommendation. In: International Joint Conference in Artificial Intelligence, vol. 21, pp. 3356–3362 (2021)
[39] Zhang, Z., Wang, B.: Prompt learning for news recommendation. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 227–237 (2023)