A First Look at LLM-Powered Generative News Recommendation

Qijiong Liu
liu@qijiong.work
The Hong Kong Polytechnic University
Hong Kong, China

Nuo Chen
pleviumtan@toki.waseda.jp
Waseda University
Tokyo, Japan

Tetsuya Sakai
tetsuyasakai@acm.org
Waseda University
Tokyo, Japan

Xiao-Ming Wu*
xiao-ming.wu@polyu.edu.hk
The Hong Kong Polytechnic University
Hong Kong, China

ABSTRACT

Personalized news recommendation systems have become essential tools for users to navigate the vast amount of online news content, yet existing news recommenders face significant challenges such as the cold-start problem, user profile modeling, and news content understanding. Previous works have typically followed an inflexible routine to address a particular challenge through model design, but are limited in their ability to understand news content and capture user interests. In this paper, we introduce GENRE, an LLM-powered generative news recommendation framework, which leverages pretrained semantic knowledge from large language models to enrich news data. Our aim is to provide a flexible and unified solution for news recommendation by moving from model design to prompt design. We showcase the use of GENRE for personalized news generation, user profiling, and news summarization. Extensive experiments with various popular recommendation models demonstrate the effectiveness of GENRE. We will publish our code and data[1] for other researchers to reproduce our work.

CCS CONCEPTS

• **Information systems** → **Personalization**; *Data mining*; **Recommender systems**.

KEYWORDS

large language model, news recommendation, data augmentation, generative information retrieval

ACM Reference Format:

Qijiong Liu, Nuo Chen, Tetsuya Sakai, and Xiao-Ming Wu. 2018. A First Look at LLM-Powered Generative News Recommendation. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

*Corresponding author.

[1] https://github.com/Jyonn/GENRE-requests

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Online news platforms, such as Google News, play a vital role in disseminating information worldwide. However, the sheer volume of articles available on these platforms can be overwhelming for users. Hence, news recommender systems have become an essential component, guiding users navigate through a vast amount of content and pinpointing articles that align with their interests.

Nonetheless, present news recommendation systems face several major challenges. One such challenge is the well-known ***cold-start problem***, a scenario where many long-tail or new users have limited browsing history, making it difficult to accurately model and understand their interests. ***User profile modeling*** poses another challenge, since user profiles consist of highly condensed information, such as geographic location or topics of interest, which are frequently withheld from datasets due to privacy concerns. Additionally, there is the unique challenge of ***news content understanding*** in the field of news recommendation. Due to limited and unaligned text information in news datasets, it can be challenging for models to capture the deeper semantics of news articles. For instance, in an article (in the MIND [48] dataset) with the title "*Here's Exactly When To Cook Every Dish For Thanksgiving Dinner*", the main idea may be "*guidance*" or "*instructions*" rather than the specific terms mentioned in the title. However, accurately identifying key concepts or themes in news articles can be challenging, which in turn affects the ability of news recommender systems to provide personalized recommendations to users.

Previous works [19, 41, 45] have proposed various recommendation models to tackle the aforementioned challenges. However, due to the limited data and knowledge available in the training dataset, these models are limited in their ability to understand news content and capture user interests. Although some methods [37] have attempted to incorporate external sources, such as knowledge graphs, their performance is often constrained by the size of the knowledge graphs.

LLM-powered generative news recommendation: a novel perspective. The advancement of large language models (LLMs), such as ChatGPT[2] or LLaMA [32], has revolutionized the field of natural language processing. The exceptional language modeling capability of LLMs enables them to understand complex patterns and relationships in language. As powerful few-shot learners, they can quickly learn the distribution of news data and incorporate relevant contextual information to improve their understanding of the

[2] https://chat.openai.com

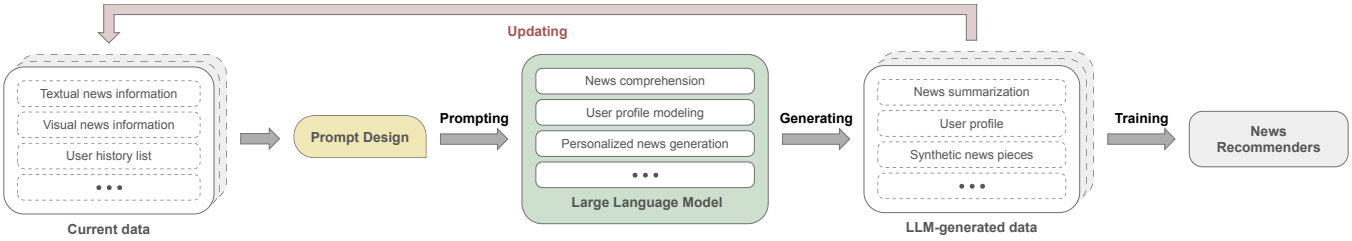


Figure 1: Our proposed LLM-powered generative news recommendation (GENRE) framework.

data. This makes LLMs a suitable tool for addressing the challenges of news recommendation systems, including the cold-start problem, user profile modeling, and news content understanding. In this work, we introduce a novel perspective for news recommendation by using LLMs to generate informative knowledge and news data such as synthetic news content tailored to cold-start users, user profiles, and refined news titles, which can be utilized to enhance the original dataset and tackle the aforementioned challenges.

Figure 1 illustrates our proposed LLM-powered **GE**nerative **N**ews **RE**commendation (**GENRE**) framework. The main idea is to utilize the available news data, such as the title, abstract, and category of each news article, to construct prompts or guidelines, which can then be fed into an LLM for producing informative news information. Due to its extensive pretrained semantic knowledge, the LLM can comprehend the underlying distribution of news data, even with very limited information provided in the original dataset, and generate enriched news data and information. These generated news data and information can be integrated back into the original dataset for the next round of knowledge generation in an iterative fashion, or utilized to train downstream news recommendation models. In this study, we explore GENRE for 1) **personalized news generation**, 2) **user profiling**, and 3) **news summarization**, to address the three challenges mentioned above.

To validate the effectiveness of our proposed GENRE framework, we perform comprehensive experiments on IM-MIND [50], a multi-modal news recommendation dataset derived from MIND [48]. We employ GPT-3.5 as the LLM and collect the generated data through API calls. Our evaluation involves four matching-based news recommendation models and four ranking-based CTR models, all of which are typical and widely used in industrial recommender systems. We observe that GENRE improves the performance of the base models significantly.

To summarize, our contributions are listed as follows:

- To our knowledge, this work is the first attempt to exploit LLMs for generative news recommendation.
- We propose GENRE, an LLM-based generative news recommendation framework. Compared to traditional methods that require designing individual models for different tasks, GENRE offers a flexible and unified solution by introducing pretrained semantic knowledge to the training data through prompt design.
- We demonstrate the effectiveness of GENRE through extensive experimentation and evaluation on three tasks: 1) personalized news generation, 2) user profiling, and 3) news summarization.

2 PRELIMINARIES

2.1 Notations and Problem Statement

Before delving into the details of our proposed method, we first introduce basic notations and formally define the news recommendation task. Let $\mathcal{N}$ be a set of news articles, where each news $n \in \mathcal{N}$ is represented by a multi-modal feature set including the title, category, and cover image. Let $\mathcal{U}$ be a set of users, where each user $u \in \mathcal{U}$ has a history of reading news articles $h^{(u)}$. Let $\mathcal{D}$ be a set of click data, where each click $d \in \mathcal{D}$ is a tuple (u, n, y) indicating whether user u clicked on news article n with label $y \in \{0, 1\}$. The task of the news recommendation is to infer the user's interest in a candidate news article.

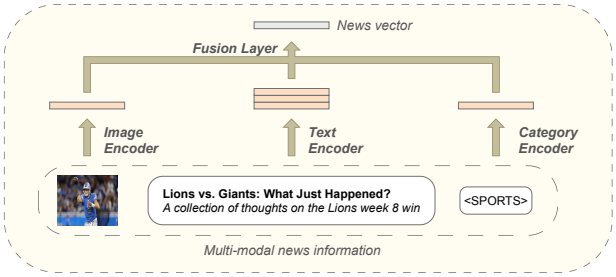
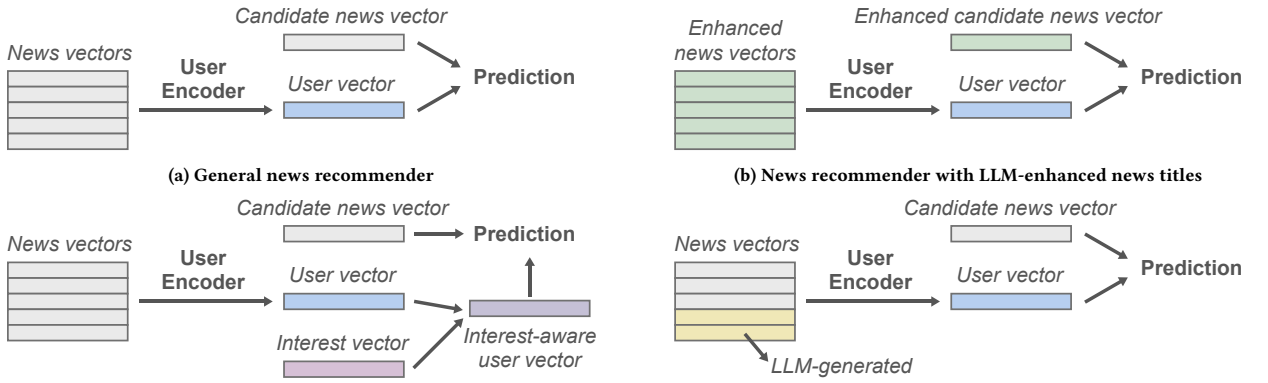


Figure 2: General multi-modal news encoder.

2.2 General News Recommendation Model

A news recommendation model generally involves three modules: a news encoder, a user encoder, and an interaction module. The news encoder, as depicted in Figure 2, is designed to encode the multi-modal features of each news article into a unified d-dimension news vector $\mathbf{v}_n$. The user encoder, as shown in Figure 3a, is designed on the top of the news encoder, generating a unified d-dimension user vector $\mathbf{v}_u$ from the sequence of browsed news vectors.

Finally, the interaction modules in **ranking models** (such as DCN [39]) and **matching models** (such as NAML [43]) have some differences. For ranking models, the click-through probability is directly calculated based on the candidate news vector $\mathbf{v}_c$ and the user vector $\mathbf{v}_u$, which is a regression problem. In contrast, for matching models, the interaction module needs to identify the positive sample that best matches the user vector $\mathbf{v}_u$ among multiple candidate news vectors $\mathbf{V}_c = [\mathbf{v}_c^{(1)}, ..., \mathbf{v}_c^{(k+1)}]$ where k is the number of negative samples, which is a classification problem.



(a) General news recommender

(b) News recommender with LLM-enhanced news titles

(c) News recommender with LLM-generated user profiles

(d) News recommender with LLM-generated personalized news pieces

Figure 3: Illustration of the downstream training of news recommenders in different scenarios.

The design of the news encoder, user encoder, and interaction module varies across different news recommendation models.

3 PROPOSED FRAMEWORK: GENRE

3.1 Overview

Figure 1 illustrate the our proposed GENRE framework for LLM-powered generative news recommendation, which consists of the following four steps. 1) Prompting: create prompts or instructions to harness the capability of a LLM for data generation for diverse objectives. 2) Generating: the LLM generates new knowledge and data based on the designed prompts. 3) Updating: use the LLM-generated data to update the current data for the next round of prompting and generation, which is optional. 4) Training: leverage the LLM-generated data to train news recommendation models.

Prompt design forms the foundation of GENRE, and the iterative generation and updating mechanism allows for an expansive and complex design space. In the following, we show examples of prompts designed under GENRE for news summarization, user profile modeling, and personalized news generation.

3.2 LLM as News Summarizer

Large language models are capable of summarizing news content into concise phrases or sentences, due to their training on vast amounts of natural language data and summarization tasks. Moreover, large language models possess remarkable skills in comprehending text, allowing them to identify noun entities like the names of individuals and locations. These entities may have appeared infrequently in the original dataset, making it challenging to learn their representations. However, large language models can associate them more effectively with knowledge learned during pre-training.

Therefore, we design a prompt for news title enhancement, as shown in Figure 4a. By providing the news title, abstract, and category as input, the large language model produces a more informative news title as output. As shown by the provided sample, the enhanced title not only summarizes the news information but also highlights the main topic of the news – "guide", which is missing from the original title.

During the training of the recommendation model, the enhanced news title will replace the original title and be used as one of the input features, together with other multi-modal features, for the news encoder (Figure 2). The green news vectors in Figure 3b represent the news vectors with the enhanced titles.

3.3 LLM as User Profiler

The user profile generally refers to their preferences and characteristics, such as age, gender, topics of interest, and geographic location. These explicit preferences often serve as important features for click-through rate (CTR) recommendation models. However, these information are usually not provided in the anonymized dataset for training recommendation models, due to privacy policies. Large language models are capable of understanding a user's reading history through their ability to model long sequences, enabling them to analyze and create an outline of the user's profile.

Hence, we design a prompt for user profiles modeling, as depicted in Figure 4b. Given a user's reading history, the large language model produces a user profile that includes his/her interested topics and regions. In this example, the LLM infers that the user may be interested in the region of Florida, based on the word "Miami" in the news. Although "Miami" may have a low occurrence in the dataset, "Florida" is more frequently represented and therefore more likely to be connected to other news or users for collaborative filtering.

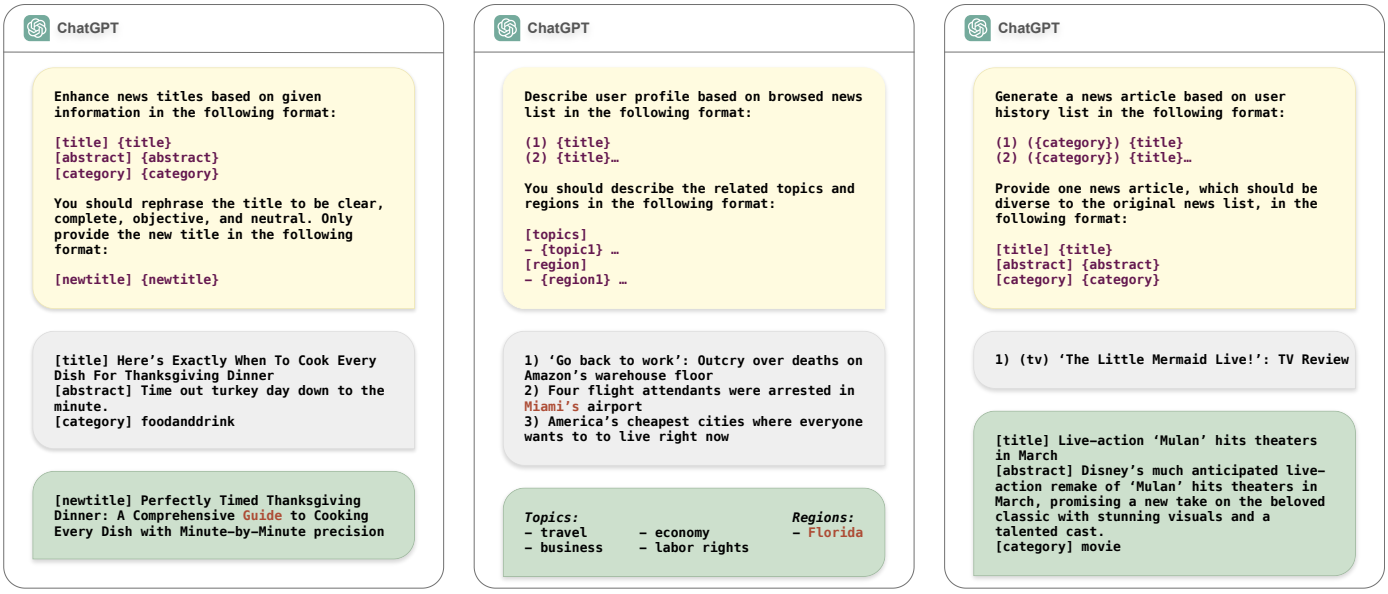
The summarized user profile will be fed into an interest fusion module which produces a interest vector $\mathbf{v}_i$ (the pink vector in Figure 3c), defined by:

$$\mathbf{v}_i = \left[\text{POOL}\left(\mathbf{E}_{\text{topics}}\right); \text{POOL}\left(\mathbf{E}_{\text{regions}}\right)\right] \in \mathbb{R}^{2\times d}, \tag{1}$$

where POOL is the average pooling operation, $\mathbf{E}_{\text{topics}}$ and $\mathbf{E}_{\text{regions}}$ are the embedding matrices of the interested topics and regions, and $[;]$ is the vector concatenation operation. The interest vector $\mathbf{v}_i$ will be combined with the user vector $\mathbf{v}_u$ (the blue vector in Figure 3c) learned by the user encoder to form the interest-aware user vector $\mathbf{v}_{iu}$ (the purple vector in Figure 3c) as follows:

$$\mathbf{v}_{iu} = \text{MLP}\left([\mathbf{v}_u; \mathbf{v}_i]\right) \in \mathbb{R}^d, \tag{2}$$

where MLP is a multi-layer perceptron with ReLU activation.



(a) News summarizer **(b) User profiler** **(c) Personalized news generator**

Figure 4: Prompts designed under our GENRE framework.

3.4 LLM as Personalized News Generator

The cold-start problem, which is well-known for its difficulties, occurs when new users[3] have limited interaction data, making it difficult for the user encoder to capture their characteristics and ultimately weakening its ability to model warm users [4]. Recent studies [7, 33] have shown that LLMs possess exceptional capabilities to learn from few examples. Hence, we propose to use an LLM to model the distribution of user-interested news given very limited user historical data. Specifically, we use it as a personalized news generator to generate synthetic news that may be of interest to new users, enhancing their historical interactions and allowing the user encoder to learn effective user representations.

The prompt displayed in Figure 4c serves as a guide for the personalized news generator, allowing the LLM to create synthetic news pieces tailored to the user's interests. The generated news pieces (indicated by the yellow news vectors in Figure 3d) are incorporated into the user historical sequence, which will be encoded and fed to the user encoder to generate the user vector.

3.5 Chain-based Generation

While we have shown several examples of "one-pass generation" (Figure 4) under our GENRE framework, it is worth noting that the design space of GENRE is vast and of a high-order complexity. As illustrated by the diagram in Figure 1, GENRE enables iterative generation and updating. The data generated by the LLM can be leveraged to enhance the quality of current data, which can subsequently be utilized in the next round of generation and prompting

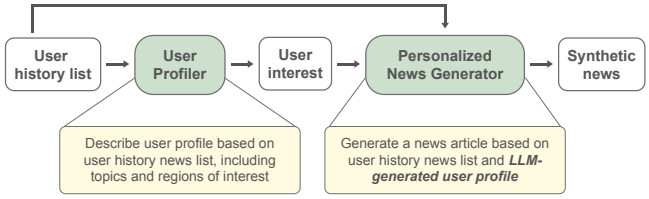


Figure 5: Chain-based personalized news generator.

in an iterative fashion. We refer to this type of generation as "chain-based generation", in contrast to "one-pass generation".

We design a chain-based personalized news generator by combining the one-pass user profiler and personalized news generator. As illustrated in Figure 5, we first use the LLM to generate the interested topics and regions of a user, which are then combined with the user history news list to prompt the LLM to generate synthetic news pieces. The user profile helps the LLM to engage in chain thinking, resulting in synthetic news that better matches the user's interests than the one-pass generator. The prompt for the chain-based generator is provided in the supplementary materials.

3.6 Downstream Training

Our GENRE framework can be applied with any news recommendation model. Existing news recommendation models mainly include matching-based models such as NAML [43], LSTUR [1], NRMS [45], and PLMNR [46], and ranking-based deep CTR models, such as BST [4], DCN [39], PNN [26], and DIN [55].

Since ranking-based models directly calculate the click-through rate, they place greater emphasis on the design of multiple feature interactions, compared to the relatively straightforward design of

[3]Following [14], we use the term "new users" to refer to users with less than 5 news articles in their reading history.
[4]We use warm user to represent the user who has browsed more than five news articles in the history.

Table 1: Statistics of the datasets.

MIND			
#news	65,238	#users	94,057
#new user	20,110	#new user ratio	0.21
#pos	347,727	#neg	8,236,715
#category	18	#subcategory	270
tokens per title	13.56	tokens per abstract	31.86
news per user	14.98	news per new user	3.19
MIND-NS			
tokens per title	16.73 (+3.17)		
MIND-UP			
#topics	1,021	#regions	117
topics per user	4.82	regions per user	0.29
MIND-NG			
#news	105,458 (+40,220)	news per new user	5.19 (+2.00)

the news encoder and user encoder (Table 2). These models are trained with the binary cross-entropy loss defined as:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{z}\sum_{i=1}^{z} y_i \log\left(\hat{y}_i\right) + \left(1 - y_i\right) \log\left(1 - \hat{y}_i\right), \tag{3}$$

where z is the batch size, y_i is the label of the i-th sample (can be 0 or 1), and $\hat{y}_i$ is the predicted probability of the i-th sample.

In contrast, matching-based models concentrate on capturing semantic information from news features and user interests. Therefore, they prioritize the design of news encoder and user encoder and use a relatively simple interaction module (Table 2). These models are trained using the cross-entropy loss:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{z}\sum_{i=1}^{z}\sum_{j=1}^{k+1} y_{i,j} \log\left(\hat{y}_{i,j}\right), \tag{4}$$

where k is the number of negative samples, $y_{i,j}$ is the label of the j-th sample in the i-th sample group (can be 0 or 1), and $\hat{y}_{i,j}$ is the predicted probability of the j-th sample in the i-th sample group.

4 EXPERIMENTS

4.1 Experimental Setup

Datasets. We conduct experiments on a large-scale real-world news recommendation dataset, MIND [48], where each news article contains features including title, abstract, category, subcategory, and cover image. The cover image of a news article is crawled from the URLs provided by IMRec [50]. In Table 1, we present the statistics of both the original dataset and the augmented versions, denoted as MIND-NS, MIND-UP, and MIND-NG, which correspond to dataset enhanced using the news summarizer, user profiler, and personalized news generator, respectively. We use OpenAI package[5] to call GPT-3.5 APIs[6] for data generation. The cost of constructing MIND-NS, MIND-UP, and MIND-NG was approximately 60 USD, 120 USD, and 40 USD, respectively. For the augmented datasets, only the attributes that are different than the orginal datasets are shown in Table 1.

[5] https://pypi.org/project/openai/
[6] https://platform.openai.com/docs/guides/chat

Table 2: Architectures of the recommendation models. Top: matching-based models. Bottom: ranking-based models.

Model	News Encoder	User Encoder	Interaction Module
NAML [43]	CNN [18]	Attention [2]	Dot Product
LSTUR [1]	CNN [18]	GRU [5]	Dot Product
NRMS [45]	Attention [34]	Attention [34]	Dot Product
PLMNR [46]	Attention [34]	Transformer [34]	Dot Product
BST [4]	Pooling	Transformer [34]	MLP
DCN [39]	Pooling	Pooling	Deep & Cross
PNN [26]	Pooling	Pooling	Inner Product
DIN [55]	Pooling	N/A	Attention [55]

Recommendation Models. We evaluate the effectiveness of our proposed GENRE framework with eight popular news recommendation models, including four matching-based models, namely NAML [43], LSTUR [1], NRMS [45], and PLMNR [46], and four ranking-based deep CTR models, namely BST [4], DCN [39], PNN [26], and DIN [55]. They follow a similar pipeline as described in subsection 2.2 and Figure 3a, but with differences in individual components, as summarized in Table 2.

Evaluation Metrics. We follow the common practice [25, 45, 46] to evaluate the effectiveness of news recommendation models with the widely used metrics, i.e., AUC [12], MRR [35] and nDCG [15]. In this work, we use nDCG@5 and nDCG@10 for evaluation, shortly denoted as N@5 and N@10, respectively.

News Features. To incorporate image information into text-based news recommendation models, we use a pretrained image encoder [28] to extract image features, which we treat as a news-specific token. We also treat the category and subcategory as special tokens that do not undergo tokenization. Then, we concatenate these features (i.e., title, image, and category) to form the input sequence for the news encoder. For the NAML model, since its original news encoder already incorporates the category information, we only concatenate the image and title features.

Implementation Details. We utilize pretrained "clip-vit-base-patch32" models [28] to extract cover image features and BertTokenizer provided by the transformers package [42] to tokenize textual features of news articles. During training, we employ Adam [17] optimizer with a learning rate of 0.001 and weight decay of 0.01. For all models, the embedding dimension is set to 64. For all matching-based models, a negative sampling ratio of 4 is specified uniformly. The number of transformer layers is set to 3 for both PLMNR [46] and BST [4]. For PLMNR, we use its best variant, PLMNR-NRMS. We use news title, category, and cover image as input features for the news encoder. Since the image embedding dimension is 512, to ensure that it matches the other features, we employ a learnable projection layer to decrease its dimension from 512 to 64. We tune the hyperparameters of all base models to attain optimal performance. We average the results of five independent runs for each model

Table 3: Effectiveness of the news summarizer (NS). ORI: training with the original data.

Matching	NAML		LSTUR		NRMS		PLMNR	
	ORI	NS	ORI	NS	ORI	NS	ORI	NS
AUC	61.75	**63.73**	61.27	**62.16**	61.71	**63.85**	62.53	**64.80**
MRR	30.60	**31.83**	29.64	**30.52**	30.20	**31.57**	30.74	**33.08**
N@5	31.35	**32.94**	30.28	**31.27**	30.98	**32.35**	31.31	**34.25**
N@10	37.85	**39.24**	36.76	**37.85**	37.42	**38.80**	38.03	**40.35**
Ranking	**BST**		**DCN**		**PNN**		**DIN**	
	ORI	NS	ORI	NS	ORI	NS	ORI	NS
AUC	61.73	**62.85**	62.63	**64.19**	61.75	**63.85**	60.95	**61.26**
MRR	29.84	**31.51**	29.73	**31.96**	29.45	**31.54**	28.13	**29.72**
N@5	30.55	**32.16**	30.52	**32.67**	29.99	**32.38**	28.77	**30.38**
N@10	37.22	**38.78**	37.12	**39.16**	36.67	**38.78**	35.42	**36.76**

Table 4: Effectiveness of the user profiler (UP). ORI: training with the original data.

Matching	NAML		LSTUR		NRMS		PLMNR	
	ORI	UP	ORI	UP	ORI	UP	ORI	UP
AUC	61.75	**62.19**	61.27	**61.81**	61.71	**61.90**	62.53	**63.31**
MRR	30.60	**30.90**	29.64	**30.39**	30.20	**30.60**	30.74	**31.58**
N@5	31.35	**31.78**	30.28	**31.00**	30.98	**31.54**	31.31	**32.65**
N@10	37.85	**38.26**	36.76	**37.46**	37.42	**37.66**	38.03	**38.87**
Ranking	**BST**		**DCN**		**PNN**		**DIN**	
	ORI	UP	ORI	UP	ORI	UP	ORI	UP
AUC	61.73	**62.67**	62.63	**63.47**	61.75	**62.34**	60.95	**62.65**
MRR	29.84	**30.75**	29.73	**29.92**	29.45	**29.67**	28.13	**30.74**
N@5	30.55	**31.63**	30.52	**30.66**	29.99	**30.46**	28.77	**31.50**
N@10	37.22	**38.01**	37.12	**37.47**	36.67	**37.07**	35.42	**38.05**

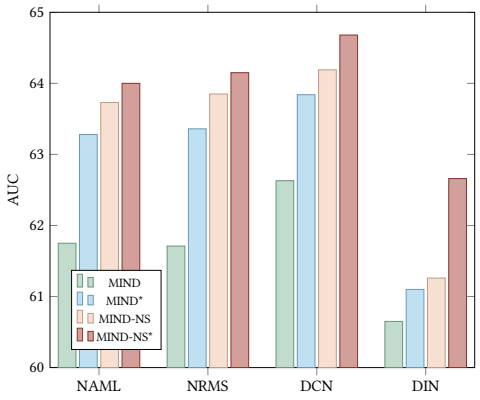


Figure 6: Influence of news features. The MIND dataset employs the original title, image, and category as inputs. The MIND-NS dataset uses the enhanced title, image, and category as inputs. The asterisk (*) represents using additional abstract and subcategory information as inputs.

and observe the p-value smaller than 0.01. All the experiments are conducted on a single NVIDIA GeForce RTX 3090 device.

4.2 LLM as News Summarizer

Table 3 presents a comparison of the performance between the original data and the data enhanced by the news summarizer for eight base models. Based on the results, we can make the following observations. **Firstly**, the improved news titles offer additional semantic information, thereby aiding the news encoder to capture the essence of news articles more effectively. **Secondly**, ranking-based models (with simple design of news encoder such as average pooling) and matching-based models (with complex design of news encoder) demonstrate comparable levels of enhancement, which could be attributed to the great importance of text keywords in comprehending news.

In the above experiments, we only utilize (enhanced) news title, category, and cover image as inputs to the news encoder. Here, we assess the impact of combining more news features. From Figure 6, the following can be summarized. **Firstly**, the inclusion of additional news features such as abstract and subcategory does lead to an improved model performance, although they are usually excluded from existing models out of efficiency concerns. **Secondly**, while MIND* has included all available news features, MIND-NS* still outperform MIND*, indicating the effectiveness of the news titles generated by GPT-3.5[7].

4.3 LLM as User Profiler

Table 4 displays a comparison of the performance between the original data and the data enhanced by the user profiler for eight base models. Based on the results, we can conclude the following. **Firstly**, the highly-summarized user interests provide valuable user knowledge to the interaction module, leading to better recommendations. **Secondly** and unsurprisingly, ranking-based models that employ a more complex interaction module show greater improvements than matching-based models that use a simple interaction module such as dot product. This is because the integration of the generated user profile (the interest vector in Figure 3c) occurs in the interaction module, and a simple combination scheme may prevent the model from fully learning the user profile features.

4.4 LLM as Personalized News Generator

Table 5 shows a comparison of the performance between the original data and the data augmented by the personalized news generator for eight base models. For each new user, two synthetic news articles are generated by the personalized news generator and appended to the user's history list. The results suggest that the

[7]It is worth noting that in recommendation systems, a one-point increase in AUC represents a significant improvement.

Table 5: Effectiveness of the personalized news generator (NG). ORI: training with the original data.

Matching	NAML		LSTUR		NRMS		PLMNR	
	ORI	NG	ORI	NG	ORI	NG	ORI	NG
AUC	61.75	**62.93**	61.27	**63.88**	61.71	**63.04**	62.53	**63.11**
MRR	30.60	**30.83**	29.64	**31.76**	30.20	**31.00**	30.74	**30.90**
N@5	31.35	**32.10**	30.28	**32.92**	30.98	**31.84**	31.31	**32.02**
N@10	37.85	**38.34**	36.76	**39.16**	37.42	**38.22**	38.03	**38.37**
Ranking	**BST**		**DCN**		**PNN**		**DIN**	
	ORI	NG	ORI	NG	ORI	NG	ORI	NG
AUC	61.73	**62.86**	62.63	**62.67**	61.75	**62.24**	60.95	**62.18**
MRR	29.84	**30.54**	29.73	**29.81**	29.45	**29.34**	28.13	**29.33**
N@5	30.55	**31.32**	30.52	**30.63**	29.99	**30.05**	28.77	**29.88**
N@10	37.22	**37.93**	37.12	**37.18**	36.67	**36.73**	35.42	**36.79**

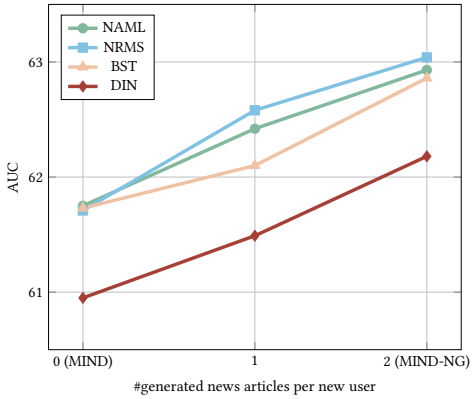


Figure 7: Influence of the number of generated news articles on the AUC metric over four base models.

generated news articles for new users align with their potential interests, leading to more reliable and accurate user representations, and consequently, an enhanced performance.

Next, we study how the number of generated news articles affects the recommendation performance. As depicted in Figure 7, we evaluate the effectiveness of utilizing 0, 1, and 2 generated news articles per new user for four base models. It can be seen that for each model, the performance improves as the number of generated news articles increases.

Additionally, we investigate the impact of the synthetic news data on two user groups, i.e., new user group and warm user group. From the results in Table 6, it can be seen that the personalized news generator improves the performance of both the new and warm user groups in most cases. This is because the user encoder struggles to capture the interests of new users due to their limited history of news consumption, which also affects its ability to model warm users. With the generated news pieces added to the browsing history of new users, the user encoder can better capture their interests, leading to a performance improvement on both groups.

Table 6: Effectiveness of the personalized news generator (NG) on the new user group and the warm user group. ORI: training with the original data. Imp.: the improvement achieved by using personalized news generator.

		New User				Warm User			
		AUC	MRR	N@5	N@10	AUC	MRR	N@5	N@10
BST	ORI	58.24	31.37	32.79	38.88	62.36	29.56	30.15	36.92
	NG	**59.36**	**31.95**	**33.40**	**39.64**	**63.50**	**30.29**	**30.94**	**37.62**
	Imp.	1.12	0.58	0.61	0.78	1.14	0.73	0.79	0.70
DIN	ORI	56.64	29.43	30.67	36.98	61.74	27.89	28.42	35.13
	NG	**59.11**	**31.07**	**32.28**	**38.69**	**62.74**	**29.02**	**29.44**	**36.44**
	Imp.	2.47	1.64	1.61	1.71	1.00	1.13	1.02	1.31
NAML	ORI	59.24	**32.82**	34.24	**40.34**	62.21	30.20	30.83	37.40
	NG	**60.21**	32.69	**34.67**	40.33	**63.43**	**30.49**	**31.64**	**37.98**
	Imp.	0.97	-0.13	0.43	-0.01	1.22	0.29	0.81	0.58
NRMS	ORI	59.49	32.75	33.99	40.09	62.12	29.74	30.43	36.93
	NG	**59.88**	**32.90**	**34.42**	**40.16**	**63.61**	**30.65**	**31.37**	**37.87**
	Imp.	0.39	0.25	0.43	0.07	1.49	0.91	0.94	0.94

4.5 Chain-based Generation

Here, we study the impact of chain-based generation on the quality of generated personalized news. Based on the results generated by the user profiler, we first remove users whose results do not contain valid topics of interest and obtain a new dataset (which accounts for **96%** of the original dataset in terms of the number of users and **97%** in terms of the number of interactions). We refer to the filtered dataset as the "Chain" dataset, in contrast to the "Full" dataset that includes all interaction data. Next, we apply chain-based generation, supplying the personalized news generator with the topics of interest generated by the user profiler, as shown in Figure 5. The performance comparison among the original dataset, the one-pass personalized news generator, and the chain-based personalized news generator (i.e., user profiler (UP) → personalized news generator (NG)) is displayed in Table 7. Based on the results, we can conclude that the knowledge learned by the user profiler improves the quality of the synthetic news produced by the personalized news generator.

4.6 Cost Conversion Rate

Finally, we investigate the cost and cost conversion rate (CCR) of different generative schemes under our GENRE framework, as presented in Table 8. We compute the average improvement in AUC compared with the original dataset for both matching and ranking models based on the results from Table 3, Table 4, and Table 5, as well as the cost conversion rate (ratio of improvement in AUC to cost of employing the GPT-3.5 API). Based on the results,

Table 7: Effectiveness of the chain-based personalized news generator (UP → NG). ORI: training with the original data. NG: training with data enhanced by the one-pass personalized news generator.

Matching	NAML				LSTUR			
	AUC	MRR	N@5	N@10	AUC	MRR	N@5	N@10
ORI	61.17	29.71	30.32	36.99	61.16	29.58	30.17	36.69
NG	62.57	30.94	32.27	38.35	62.70	30.87	31.72	38.18
UP→NG	**63.61**	**31.58**	**32.63**	**39.07**	**63.57**	**31.43**	**32.62**	**39.01**
Ranking	**BST**				**DCN**			
	AUC	MRR	N@5	N@10	AUC	MRR	N@5	N@10
ORI	61.20	29.34	29.78	36.54	62.00	29.36	30.22	36.74
NG	62.45	30.20	31.03	37.49	62.65	**29.87**	**30.74**	**37.23**
UP→NG	**63.28**	**31.49**	**32.45**	**38.84**	**63.05**	29.79	30.61	**37.23**

Table 8: Comparison of the cost and cost conversion rate (CCR) of different generative schemes. Imp.: the average improvement in AUC compared with the original dataset. CCR: the ratio of improvement to cost. Chain dataset: the filtered dataset for chain-based generation with 97% interaction data as described in subsection 4.5. Note that the cost of UP → NG is calculated by $120 \times 0.21 + 60$, where 120 is the cost of UP, 0.21 is the new user ratio, and 60 is the cost of chain-based NG.

			Matching		Ranking	
	Dataset	Cost (USD)	Imp.	CCR (%)	Imp.	CCR (%)
NS	Full	60	**1.82**	3.03	**1.27**	**2.11**
UP	Full	120	0.49	0.41	1.02	0.85
NG	Full	40	1.42	**3.55**	0.72	1.80
NG	Chain	40	1.47	**3.68**	0.95	**2.38**
UP→NG	Chain	85	**2.43**	2.86	**1.57**	1.85

we can conclude the following. **Firstly**, with the full dataset, the personalized news generator (NG) has the best CCR for matching-based models, and the news summarizer (NS) has the best CCR for ranking-based models. **Secondly**, the user profiler (UP) has the worst CCR, since the extensive length of a user's browsing history results in a high token count per request, leading to increased cost for the user profiler. **Thirdly**, chain-based generation achieves a higher improvement compared to one-pass generation (with the chain dataset), but its CCR decreases due to the use of the expensive user profiler.

5 RELATED WORKS

5.1 Generative Models for Recommendation

Over past few years, generative models have achieved great success in various fields such as natural language processing [6] and computer vision [30], and have also been explored for recommendation. Examples include generative adversarial networks [3, 38, 51, 52], variational auto-encoders [9, 21, 23, 52], and diffusion models [22, 36, 40].

The recent advancement of large language models, particularly ChatGPT, has triggered a new wave of interest, resulting in the development of diverse applications across multiple domains [7, 27, 49]. There have been several studies attempting to use ChatGPT for rating prediction and direct recommendation through in-context learning (in contrast of involving LLMs in training as part of the model). [8, 13, 16, 24]. There have also been studies applying LLMs to conversational recommender systems, where users provide natural language instructions to receive recommendation results [53]. In this paper, we make the first attempt to introduce LLMs for generative news recommendation.

5.2 News Recommendation

In the past few years, news recommendation has gained significant attention and has been widely studied in both industry and academia. To better capture textual knowledge and user preferences, several models based on deep neural networks have been proposed [1, 43–45]. These models use various techniques such as convolutional neural networks (CNNs) [18], recurrent neural networks (RNNs) [5, 11], and attention mechanisms [34] as news or user encoders. Despite their effectiveness, these end-to-end models have limited semantic comprehension abilities.

Recently, there has been a surge of interest in using pretrained language models (PLMs) such as BERT [10] and GPT [29] in news recommendation systems, owing to the powerful transformer-based architectures and the availability of large-scale pretraining data. These models have shown promising results in news recommendation [25, 46, 47, 54]. The emergence of large language models (LLMs) such as ChatGPT and LLaMa [32] has further opened up the possibility of leveraging rich general knowledge to enhance the efficacy of recommender systems. Very recently, there has been a few attempts to utilize LLMs for personalization [31] and product recommendation [20]. LaMP [31] exploits LLMs to develop a personalized benchmark for NLP tasks. GPT4Rec [20] uses LLMs to generate hypothetical search queries and retrieves items for recommendation by searching for these queries. [24] points out that directly employing LLMs as a recommender system has shown negative results [24]. The use of LLMs for news recommendation remains understudied.

Due to the large size of LLMs, it is inefficient to use them as news encoders in both the training and inference stages. In this work, we take the first step towards LLM-powered generative news recommendation by proposing a general framework that leverages the pre-trained knowledge in LLMs to enhance the training data from various aspects and improve the performance of news recommendation models.

6 CONCLUSION

Our work addresses the limitations of news recommendation systems and offers a new approach that leverages LLMs to enhance their performance. Our findings indicate that integrating the general knowledge of LLMs into recommendation systems can lead to substantial improvements, which has important implications for online news platforms. Our framework GENRE can be applied to other domains beyond news recommendation. We hope our work will encourage further research and contribute to the development of more efficient and effective recommendation systems.

REFERENCES

[1] Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural News Recommendation with Long- and Short-term User Representations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 336–345.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[3] Homanga Bharadhwaj, Homin Park, and Brian Y. Lim. 2018. RecGAN: Recurrent Generative Adversarial Networks for Recommendation Systems. In *Proceedings of the 12th ACM Conference on Recommender Systems* (Vancouver, British Columbia, Canada) *(RecSys '18)*. Association for Computing Machinery, New York, NY, USA, 372–376. https://doi.org/10.1145/3240323.3240383

[4] Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. 2019. Behavior sequence transformer for e-commerce recommendation in alibaba. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*. 1–4.

[5] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[6] Danilo Croce, Giuseppe Castellucci, and Roberto Basili. 2020. GAN-BERT: Generative adversarial learning for robust text classification with a bunch of labeled examples. In *Proceedings of the 58th annual meeting of the association for computational linguistics*. 2114–2119.

[7] Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Lichao Sun, Quanzheng Li, Dinggang Shen, Tianming Liu, and Xiang Li. 2023. AugGPT: Leveraging ChatGPT for Text Data Augmentation. arXiv:2302.13007 [cs.CL]

[8] Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. 2023. Uncovering ChatGPT's Capabilities in Recommender Systems. arXiv:2305.02182 [cs.IR]

[9] Romain Deffayet, Thibaut Thonet, Jean-Michel Renders, and Maarten de Rijke. 2023. Generative Slate Recommendation with Reinforcement Learning *(WSDM '23)*. Association for Computing Machinery, New York, NY, USA, 580–588. https://doi.org/10.1145/3539597.3570412

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

[11] Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science* 14, 2 (1990), 179–211.

[12] Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern recognition letters* 27, 8 (2006), 861–874.

[13] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chat-REC: Towards Interactive and Explainable LLMs-Augmented Recommender System. arXiv:2303.14524 [cs.IR]

[14] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*. 507–517.

[15] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[16] Wang-Cheng Kang, Jianmo Ni, Nikhil Mehta, Maheswaran Sathiamoorthy, Lichan Hong, Ed Chi, and Derek Zhiyuan Cheng. 2023. Do LLMs Understand User Preferences? Evaluating LLMs On User Rating Prediction. arXiv:2305.06474 [cs.IR]

[17] Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations* (2015).

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2017. Imagenet classification with deep convolutional neural networks. *Commun. ACM* 60, 6 (2017), 84–90.

[19] Won-Jo Lee, Kyo-Joong Oh, Chae-Gyun Lim, and Ho-Jin Choi. 2014. User profile extraction from Twitter for personalized news recommendation. In *16th International conference on advanced communication technology*. IEEE, 779–783.

[20] Jinming Li, Wentao Zhang, Tian Wang, Guanglei Xiong, Alan Lu, and Gerard Medioni. 2023. GPT4Rec: A Generative Framework for Personalized Recommendation and User Interests Interpretation. arXiv:2304.03879 [cs.IR]

[21] Xiaopeng Li and James She. 2017. Collaborative Variational Autoencoder for Recommender Systems. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Halifax, NS, Canada) *(KDD '17)*. Association for Computing Machinery, New York, NY, USA, 305–314. https://doi.org/10.1145/3097983.3098077

[22] Zihao Li, Aixin Sun, and Chenliang Li. 2023. DiffuRec: A Diffusion Model for Sequential Recommendation. arXiv:2304.00686 [cs.IR]

[23] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *Proceedings of the 2018 World Wide Web Conference* (Lyon, France) *(WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 689–698. https://doi.org/10.1145/3178876.3186150

[24] Junling Liu, Chao Liu, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is ChatGPT a Good Recommender? A Preliminary Study. *arXiv preprint arXiv:2304.10149* (2023).

[25] Qijiong Liu, Jieming Zhu, Quanyu Dai, and Xiaoming Wu. 2022. Boosting Deep CTR Prediction with a Plug-and-Play Pre-trainer for News Recommendation. In *Proceedings of the 29th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 2823–2833. https://aclanthology.org/2022.coling-1.249

[26] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-based neural networks for user response prediction. In *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 1149–1154.

[27] Basit Qureshi. 2023. Exploring the Use of ChatGPT as a Tool for Learning and Assessment in Undergraduate Computer Science Curriculum: Opportunities and Challenges. arXiv:2304.11214 [cs.CY]

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[29] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. (2018).

[30] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. 2019. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems* 32 (2019).

[31] Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2023. LaMP: When Large Language Models Meet Personalization. arXiv:2304.11406 [cs.CL]

[32] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[33] Solomon Ubani, Suleyman Olcay Polat, and Rodney Nielsen. 2023. ZeroShotDataAug: Generating and Augmenting Training Data with ChatGPT. arXiv:2304.14334 [cs.AI]

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

[35] Ellen M Voorhees et al. 1999. The trec-8 question answering track report.. In *Trec*, Vol. 99. 77–82.

[36] Joojo Walker, Ting Zhong, Fengli Zhang, Qiang Gao, and Fan Zhou. 2022. Recommendation Via Collaborative Diffusion Generative Model. In *Knowledge Science, Engineering and Management: 15th International Conference, KSEM 2022, Singapore, August 6–8, 2022, Proceedings, Part III* (Singapore, Singapore). Springer-Verlag, Berlin, Heidelberg, 593–605. https://doi.org/10.1007/978-3-031-10989-8_47

[37] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 world wide web conference*. 1835–1844.

[38] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. IRGAN: A Minimax Game for Unifying Generative and Discriminative Information Retrieval Models. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Shinjuku, Tokyo, Japan) *(SIGIR '17)*. Association for Computing Machinery, New York, NY, USA, 515–524. https://doi.org/10.1145/3077136.3080786

[39] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & Cross Network for Ad Click Predictions. In *Proceedings of the ADKDD'17* (Halifax, NS, Canada) *(ADKDD'17)*. Association for Computing Machinery, New York, NY, USA, Article 12, 7 pages.

[40] Wenjie Wang, Yiyan Xu, Fuli Feng, Xinyu Lin, Xiangnan He, and Tat-Seng Chua. 2023. Diffusion Recommender Model. arXiv:2304.04971 [cs.IR]

[41] Yinwei Wei, Xiang Wang, Qi Li, Liqiang Nie, Yan Li, Xuanping Li, and Tat-Seng Chua. 2021. Contrastive learning for cold-start recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia*. 5382–5390.
[42] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*. 38–45.
[43] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, et al. 2019. Neural news recommendation with attentive multi-view learning. In *International Joint Conferences on Artificial Intelligence*.
[44] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. NPA: Neural News Recommendation with Personalized Attention. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 2576–2584. https://doi.org/10.1145/3292500.3330665
[45] Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019. 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)*. 6389–6394.
[46] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Empowering news recommendation with pre-trained language models. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1652–1656.
[47] Chuhan Wu, Fangzhao Wu, Yang Yu, Tao Qi, Yongfeng Huang, and Qi Liu. 2021. NewsBERT: Distilling Pre-trained Language Model for Intelligent News Application. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 3285–3295. https://doi.org/10.18653/v1/2021.findings-emnlp.280
[48] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, et al. 2020. Mind: A large-scale dataset for news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 3597–3606.
[49] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. BloombergGPT: A Large Language Model for Finance. arXiv:2303.17564 [cs.LG]
[50] Jiahao Xun, Shengyu Zhang, Zhou Zhao, Jieming Zhu, Qi Zhang, Jingjie Li, Xiuqiang He, Xiaofei He, Tat-Seng Chua, and Fei Wu. 2021. Why do we click: visual impression-aware news recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia*. 3881–3890.
[51] Ruiping Yin, Kan Li, Jie Lu, and Guangquan Zhang. 2019. RsyGAN: Generative Adversarial Network for Recommender Systems. In *2019 International Joint Conference on Neural Networks (IJCNN)*. 1–7. https://doi.org/10.1109/IJCNN.2019.8851727
[52] Xianwen Yu, Xiaoning Zhang, Yang Cao, and Min Xia. 2019. VAEGAN: A Collaborative Filtering Framework based on Adversarial Variational Autoencoders. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 4206–4212. https://doi.org/10.24963/ijcai.2019/584
[53] Junjie Zhang, Ruobing Xie, Yupeng Hou, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2023. Recommendation as Instruction Following: A Large Language Model Empowered Recommendation Approach. arXiv:2305.07001 [cs.IR]
[54] Qi Zhang, Jingjie Li, Qinglin Jia, Chuyuan Wang, et al. 2021. UNBERT: User-News Matching BERT for News Recommendation. In *International Joint Conferences on Artificial Intelligence*.
[55] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.

A ADDITIONAL RESULTS ON CHAIN-BASED GENERATION

Here, we provide additional results on chain-based generation (Section 4.5 in the main text), by conducting experiments with the other four recommendation models. The same observation can be made – the knowledge learned by the user profiler improves the quality of the synthetic news produced by the personalized news generator.

We also study the effect of the number of generated news articles on recommendation performance in **??**, which shows the performance of the chain-based personalized news generator improves as the number of generated news articles increases.

B COMBINATION OF THE THREE SCHEMES

Here, we investigate the combination of the news summarizer, user profiler, and personalized news generator. We first use the news summarizer to enhance the news title of each news article. Then, we use the user profiler to generate user profiles for all users. Next, for users with valid profiles, we use the chain-based personalized news generator to create two synthetic news articles to extend their history lists. If the user profile is empty, we leave their history lists unchanged rather than removing these users as we did for the "Chain" dataset. We use "ALL" to denote such combination. The results in Table 10 show that the combination of the three generative schemes (i.e., "ALL") achieve the best performance for all recommendation models in most cases, significantly outperforming training with the original data (ORI).

Table 9: Effectiveness of the chain-based personalized news generator (UP → NG). ORI: training with the original data. NG: training with data enhanced by the one-pass personalized news generator.

Matching	NRMS				PLMNR			
	AUC	MRR	N@5	N@10	AUC	MRR	N@5	N@10
ORI	61.88	30.63	31.38	37.76	62.27	31.07	31.98	38.31
NG	62.40	30.68	31.41	37.82	63.00	31.03	32.10	38.41
UP→NG	**62.95**	**32.00**	**32.80**	**39.00**	**64.02**	**31.98**	**33.25**	**39.40**
Ranking	**PNN**				**DIN**			
	AUC	MRR	N@5	N@10	AUC	MRR	N@5	N@10
ORI	61.68	29.21	29.72	36.44	60.80	28.45	28.87	35.60
NG	62.25	29.51	30.12	36.87	62.87	30.42	30.88	37.70
UP→NG	**63.63**	**30.85**	**31.14**	**38.69**	**63.53**	**30.76**	**31.21**	**38.13**

Table 10: Performance comparison among the one-pass news summarizer (NS), one-pass user profiler (UP), one-pass personalized news generator (NG), chain-based personalized news generator (UP→NG), and ALL that combines the news title generated by the one-pass news summarizer and synthetic news generated by the chain-based personalized news generator. ORI: training with the original data. Chain dataset: the filtered dataset produced by chain-based generation with 97% interaction data as described in Section 4.5 of the main text. Except for the results of ALL, the other results are copied from Table 3, 4, 5, and 7 in the main text and Table 9 in the supplementary material for ease of comparison.

Matching	Dataset	NAML				LSTUR				NRMS				PLMNR			
		AUC	MRR	N@5	N@10	AUC	MRR	N@5	N@10	AUC	MRR	N@5	N@10	AUC	MRR	N@5	N@10
ORI	Full	61.75	30.60	31.35	37.85	61.27	29.64	30.28	36.76	61.71	30.20	30.98	37.42	62.53	30.74	31.31	38.03
NS	Full	63.73	31.83	32.94	39.24	62.16	30.52	31.27	37.85	**63.85**	31.57	32.35	38.80	64.80	**33.08**	34.25	40.35
UP	Full	62.19	30.90	31.78	38.26	61.81	30.39	31.00	37.46	61.90	30.60	31.54	37.66	63.31	31.58	32.65	38.87
NG	Full	62.93	30.83	32.10	38.34	63.88	31.76	32.92	39.16	63.04	31.00	31.84	38.22	63.11	30.90	32.02	38.37
UP→NG	Chain	63.61	31.58	32.63	39.07	63.57	31.43	32.62	39.01	62.95	32.00	32.80	39.00	64.02	31.98	33.25	39.40
ALL	Full	**63.88**	**32.17**	**33.14**	**39.37**	**64.04**	**32.40**	**33.30**	**39.47**	63.71	**32.14**	**33.11**	**39.43**	**65.13**	32.98	**34.30**	**40.49**
Ranking	**Dataset**	**BST**				**DCN**				**PNN**				**DIN**			
		AUC	**MRR**	**N@5**	**N@10**	**AUC**	**MRR**	**N@5**	**N@10**	**AUC**	**MRR**	**N@5**	**N@10**	**AUC**	**MRR**	**N@5**	**N@10**
ORI	Full	61.73	29.84	30.55	37.22	62.63	29.73	30.52	37.12	61.75	29.45	29.99	36.67	60.95	28.13	28.77	35.42
NS	Full	62.85	31.51	32.16	38.78	64.19	31.96	32.67	39.16	63.85	31.54	32.38	38.78	61.26	29.72	30.38	36.76
UP	Full	62.67	30.75	31.63	38.01	63.47	29.92	30.66	37.47	62.34	29.67	30.46	37.07	62.65	30.74	31.50	38.05
NG	Full	62.86	30.54	31.32	37.93	62.67	29.81	30.63	37.18	62.24	29.34	30.05	36.73	62.18	29.33	29.88	36.79
UP→NG	Chain	63.28	31.49	32.45	38.84	63.05	29.79	30.61	37.23	63.63	30.85	31.14	38.69	63.53	30.76	31.21	38.13
ALL	Full	**63.94**	**32.05**	**33.09**	**39.41**	**65.77**	**32.86**	**34.10**	**40.48**	**65.49**	**32.78**	**33.81**	**40.19**	**63.80**	**31.68**	**32.57**	**39.08**