RESEARCH-ARTICLE

ONCE: Boosting Content-based Recommendation with Both Open- and Closed-source Large Language Models

QIJIONG LIU, The Hong Kong Polytechnic University, Hong Kong, Hong Kong, Hong Kong

NUO CHEN, Waseda University, Tokyo, Japan

TETSUYA SAKAI, Waseda University, Tokyo, Japan

XIAOMING WU, The Hong Kong Polytechnic University, Hong Kong, Hong Kong, Hong Kong

Open Access Support provided by:

Waseda University

The Hong Kong Polytechnic University

Published: 04 March 2024

Citation in BibTeX format

WSDM '24: The 17th ACM International Conference on Web Search and Data Mining
March 4 - 8, 2024
Merida, Mexico

Conference Sponsors:
SIGKDD
SIGMOD
SIGIR
SIGWEB

ONCE: Boosting Content-based Recommendation with Both Open- and Closed-source Large Language Models

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 content-based recommender systems have become indispensable tools for users to navigate through the vast amount of content available on platforms like daily news websites and book recommendation services. However, existing recommenders face significant challenges in understanding the content of items. Large language models (LLMs), which possess deep semantic comprehension and extensive knowledge from pretraining, have proven to be effective in various natural language processing tasks. In this study, we explore the potential of leveraging both open- and closed-source LLMs to enhance content-based recommendation. With open-source LLMs, we utilize their deep layers as content encoders, enriching the representation of content at the embedding level. For closed-source LLMs, we employ prompting techniques to enrich the training data at the token level. Through comprehensive experiments, we demonstrate the high effectiveness of both types of LLMs and show the synergistic relationship between them. Notably, we observed a significant relative improvement of up to 19.32% compared to existing state-of-the-art recommendation models. These findings highlight the immense potential of both open- and closed-source of LLMs in enhancing content-based recommendation systems. We have made our code and LLM-generated data available[1] for other researchers to reproduce our results.

CCS CONCEPTS

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

KEYWORDS

large language model, content-based recommendation

*Corresponding author.
[1]https://github.com/Jyonn/ONCE

WSDM '24, March 4–8, 2024, Merida, Mexico

ACM ISBN 979-8-4007-0371-3/24/03…$15.00
https://doi.org/10.1145/3616855.3635845

ACM Reference Format:
Qijiong Liu, Nuo Chen, Tetsuya Sakai, and Xiao-Ming Wu. 2024. 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 (WSDM '24), March 4–8, 2024, Merida, Mexico.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3616855.3635845

1 INTRODUCTION

Content-based recommender systems [48] analyze the content and properties of items (e.g., articles, movies, books, or products) to deliver relevant and personalized recommendations to users. Some instances of such systems are Google News[2], which offers recommendations for news articles, and Goodreads[3], which provides recommendations for books. With the rapid expansion of digital content, it becomes increasingly essential to improve content-based recommendation techniques in order to meet users' expectations for precise and pertinent recommendations.

The core component of content-based recommender systems is the *content encoder*, which is used for encoding the textual information of items in order to capture semantic features. In the past, recommendation models [1, 40, 42] commonly utilized convolutional neural networks (CNNs) as content encoders, typically initialized with pre-trained word representations such as GloVe [26]. In recent years, recommendation methods [43] have made use of pretrained language models (PLMs) based on the Transformer architecture [34] to extract more comprehensive semantic information. Despite these advancements, existing methods still struggle to fully comprehend the content of items.

To illustrate the limitations of previous content encoders, we present an example in Figure 2. We chose three books from the Goodreads dataset: "*The Lion King*", a novel adapted from a Disney animated movie, and the historical fantasy novels "*The Lions of Al-Rassan*" and "*The Summer Tree*", both authored by Guy Gavriel Kay, belonging to a distinct category. We use different encoders to encoder the titles of these books and visualize their relative similarities in the embedding space. The results reveal that early content encoders relying on pretrained word embeddings struggle at the *word level*, failing to recognize crucial terms like "Al-Rassan" and resulting in erroneously high similarity between "The Lion King" and

[2] https://news.google.com/
[3] https://www.goodreads.com/

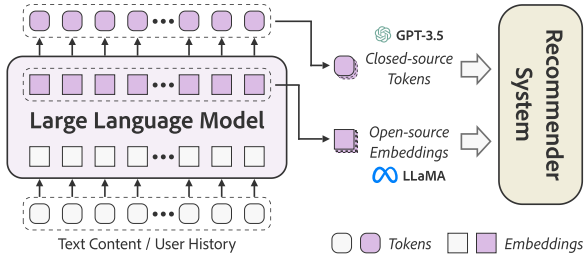


Figure 1: Employing two different types of LLMs for content-based recommendations.

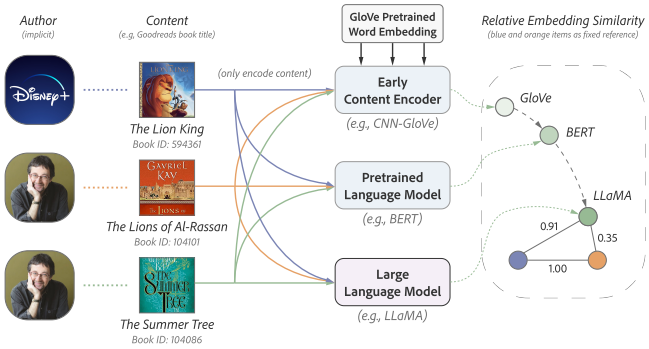


Figure 2: Comparison of content encoders used for content-based recommendation. To illustrate the similarity among the three books, we employ relative distance to align the three different embedding spaces. First, we compute the cosine similarities $s_{i,j}$ between each pair of books. Then, we calculate their relative distances using $d_{i,j} = (1 - s_{i,j})/(1 - s_{blue,orange})$, i.e., by fixing the similarity between "*The Lion King*" and "*The Lions of Al-Rassan*" as 1. This approach allows for a direct comparison of their similarities across the three distinct embedding spaces. It's important to note that a shorter distance indicates a greater similarity.

"The Lions of Al-Rassan". Similarly, small-scale pretrained language models (PLMs) face challenges at the *content level*. Constrained by their pretraining data lacking relevant knowledge and their limited representation dimensions (e.g., 768), they are unable to fully grasp the content of "The Lions of Al-Rassan" and "The Summer Tree" and accurately perceive their similarity, leading to outcomes similar to the early content encoders.

Such limitations can be overcome by the emerging large language models (LLMs), with the likes of closed-source ChatGPT[4] and open-source LLaMA [30] leading the way. These models possess billions of parameters and are trained on datasets containing trillions of tokens. With each token represented in thousands of dimensions, they can store an extensive amount of information. In contrast to small PLMs like BERT, these LLMs demonstrate remarkable "emergent abilities" [39] in terms of advanced language comprehension and generation capabilities, making it possible to deliver more contextually relevant and personalized recommendations. When we use ChatGPT to inquire about a book, it showcases its enriched knowledge at the content level by providing detailed information such as the author, publication date, and subject matter. In Figure 2, we initially prompted LLaMA to generate concise descriptions of the three books solely based on the title information, and then employed it to encode these descriptions and obtain the corresponding representations[5]. The results clearly indicate that the representations generated by LLaMA accurately reflect the similarity in content between the three books: the similarity between "*The Lions of Al-Rassan*" and "*The Summer Tree*" is higher than their similarity with "*The Lion King*".

In this paper, we investigate the possibility of enhancing content-based recommendation by leveraging both **O**pe**N**- and **C**los**E**d-source (**ONCE**) LLMs. As depicted in Figure 1, our approach ONCE adopts different strategies for each type of LLMs. For open-source LLMs like LLaMA, we employ a ***di**scriminative* **re**commendation approach named **DIRE**, reminiscent of the PLM-NR [43] method, by replacing the original content encoder with the LLM. This enables us to extract content representations and fine-tune the model specifically for recommendation tasks, ultimately enhancing user modeling and content understanding. Conversely, for closed-source LLMs like GPT-3.5, where we only have access to token outputs, we propose a ***gen**erative* **re**commendation approach named **GENRE**. By devising various prompting strategies, we enrich the available training data and acquire more informative textual and user features, which contribute to improved performance in downstream recommendation tasks.

We conducted extensive evaluations using two well-established content recommendation benchmarks: MIND [45] and Goodreads [36]. Our main objective was to thoroughly assess the impact of both open-source and closed-source LLMs on content-based recommendation models, focusing on recommendation quality and training efficiency. The results of our study demonstrate that both open- and closed-source LLMs are highly effective, especially the former. Through the process of finetuning LLaMA, we consistently observed enhancements of more than 10 percentage points compared to existing state-of-the-art recommendation models. Additionally, we discovered a complementary relationship between open- and closed-source LLMs. Specifically, the enriched data generated by ChatGPT substantially accelerated the efficiency of finetuning LLaMA while simultaneously enhancing recommendation quality. Our findings highlight the immense potential of both types of LLMs in enhancing content-based recommendation systems.

2 OVERVIEW

2.1 Content-based Recommendation

Before delving into the details of our proposed method, we first introduce basic notations and formally define the content-based recommendation task. Let $\mathcal{N}$ represent the set of contents, where each content $n \in \mathcal{N}$ is characterized by a diverse feature set, such as title, category, or description, in various recommendation scenarios. Similarly, let $\mathcal{U}$ denote the set of users, where each user $u \in \mathcal{U}$ maintains a history of browsed content denoted as $h^{(u)}$. Additionally, $\mathcal{D}$ corresponds to the set of click data, with each click $d \in \mathcal{D}$ represented as a tuple (u, n, y), indicating whether user u

[4] https://chat.openai.com
[5] It is important to note that in our experiments in Section 5, we used LLaMA as a content encoder without any prompting.

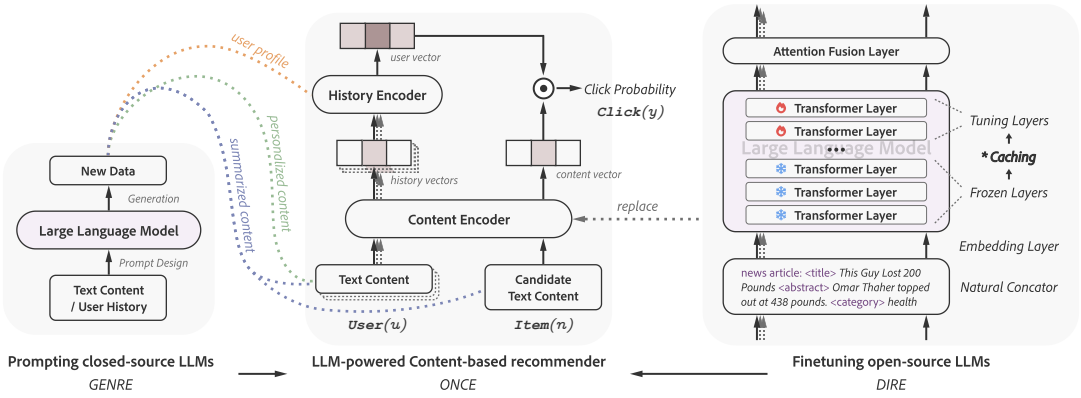


Figure 3: An overview of our proposed ONCE framework, designed to enhance content-based recommendations by finetuning open-source LLMs (DIRE) and employing prompts for closed-source LLMs (GENRE).

clicked on content n with label $y \in 0, 1$. The objective of content-based recommendation is to infer the user's interest in a candidate content.

A content-based recommendation model typically consists of three core modules: a content encoder, a history encoder, and an interaction module. The content encoder is responsible for encoding the multiple features of each content, consolidating them into a unified d-dimensional content vector $\mathbf{v}_n$. On top of the content encoder, the history encoder generates a unified d-dimensional user vector $\mathbf{v}_u$ based on the sequence of browsed content vectors.

Finally, the interaction module aims to identify the positive sample that best aligns with the user vector $\mathbf{v}_u$ among multiple candidate content vectors $\mathbf{V}_c = [\mathbf{v}_c^{(1)}, ..., \mathbf{v}_c^{(k+1)}]$, where k represents the number of negative samples. This process can be viewed as a classification problem.

2.2 Enhancing Content-based Recommendation with Open- and Closed-source LLMs (ONCE)

Large language models, endowed with deep semantic understanding and comprehensive knowledge acquired from pretraining, have exhibited proficiency across a multitude of natural language processing tasks. In this paper, we introduce the ONCE framework, which leverages both open-source and closed-source LLMs to enhance content-based recommendations. As highlighted in Touvron et al. [31], there remains a discernible gap between open-source models, encompassing approximately 10 billion parameters, and the closed-source GPT-3.5, an expansive entity boasting over 175 billion parameters. Our ONCE framework capitalizes on the strengths of both types and constructs a more robust recommendation system. We initiate the process by utilizing the closed-source LLM through prompting, enhancing the dataset from various perspectives, following our designed generative recomendation framework (GENRE). This infusion of external knowledge, a facet not readily accessible to open-source models, ensues. Subsequently, we propose a discriminative recommendation framework (DIRE) to harness the deep layers of the open-source LLM as content encoders, thereby amplifying content representations.

3 DIRE: FINETUNING OPEN-SOURCE LLMS

Integrating open-source language models as content encoders is a straightforward and widely adopted method in content-based recommendation [25, 43]. Notably, PLM-NR [43] employs small-scale pretrained language models (PLMs, e.g., BERT [9]) to replace original news encoders and finetunes on the recommendation task.

The success of this approach relies on two factors: 1) the knowledge inherent in the pretrained language models (including model size and pretraining data quality), and 2) the finetuning strategy. As discussed earlier, we have already highlighted the advantages of large language models in content understanding and user modeling, addressing the first factor. In this section, we propose discriminative recommendation framework, namely **DIRE**, and explore how to leverage open-source large language models to further enhance recommendation performance by considering the second factor.

3.1 Network Architecture

As depicted in Figure 3, we seamlessly incorporate the open-source large language model and an attention fusion layer into the content-based recommendation framework.

Embedding Layer. In contrast to the approach taken by smaller-scale PLMs like BERT, which utilize specific tokens (e.g., ⟨cls⟩, ⟨sep⟩) to segment distinct fields, we adopt natural language templates for concatenation. For instance, consider a news content n containing attributes such as title, abstract, and category features. As illustrated in Figure 3, We introduce the label "*news article:*" at the outset of the sequence, while each feature is prefixed with "⟨*feature*⟩". This procedure transforms the multi-field content into a cohesive individual sequence $\mathbf{s}$ of length l. We refer to this technique as the "**Natural Concator**". Following this, we make use of pretrained token embeddings provided by the LLM to map discrete text sequence into a continuous embedding space of dimensionality d^n, denoted as:

$$\mathbf{E}^0 = EmbeddingLayer\,(\mathbf{s}) \in \mathbb{R}^{l \times d^n}. \tag{1}$$

Transformer Decoder. The design of the LLM (or LLaMA) is based on the Transformer architecture [34], incorporating multiple tiers of Transformer Layers. This configuration is intricately interconnected, with the output hidden state from each layer feeding

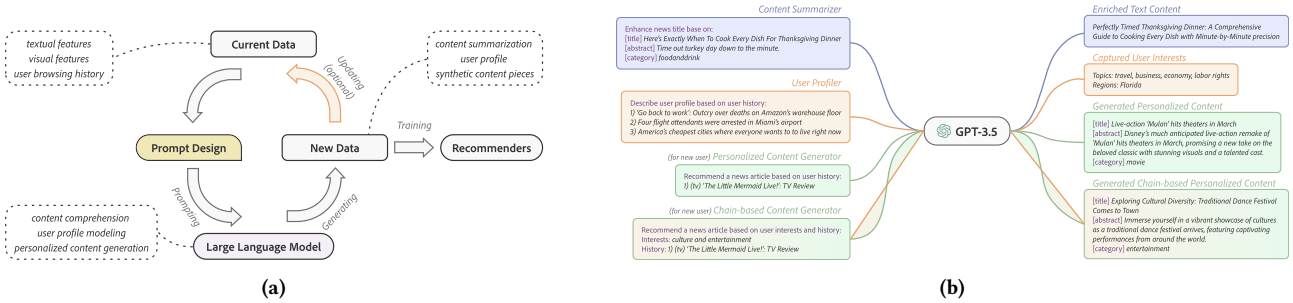


Figure 4: (a) Our proposed GENRE framework. (b) Prompting closed-source GPT-3.5 as data augmenter.

into the input of the next layer, denoted as:

$$\mathbf{E^i} = TransformerLayer\left(\mathbf{E^{i-1}}\right) \in \mathbb{R}^{l \times d^n}, i \in \{1, ..., H\}, \quad (2)$$

where H represents the number of Transformer Layers.

Attention Fusion Layer. To combine the sequential hidden states from the last layer into a single cohesive content representation, we employ the attention fusion layer, following a similar approach as used in PLM-NR [43]. Specifically, we begin by mapping the high-dimensional hidden states from a large space of dimensionality d^n to a smaller d-dimensional space (where $d^n \gg d$), defined by:

$$\mathbf{Z} = \mathbf{E^i W} + \mathbf{b} \in \mathbb{R}^{l \times d}, \quad (3)$$

where $\mathbf{W} \in \mathbb{R}^{d^n \times d}$ and $\mathbf{b} \in \mathbb{R}^d$ are the learnable parameters of the linear transformation. Next, we utilize the additive attention mechanism [2] to further condense the reduced representation into a unified representation $\mathbf{z}$, defined by:

$$\mathbf{z} = Attention\left(\mathbf{Z}\right) \in \mathbb{R}^d, \quad (4)$$

which will be fed into the user modeling module or interaction module for further personalized recommendation.

3.2 Finetuning Strategy

Partial Freezing and Caching. Running large language models incurs significant computational demands due to their expansive collection of transformer layers and associated parameters. Given that the lower layers of the LLM tend to possess a more generalized and less task-specific nature, we opt to keep these layer parameters fixed. Instead, we exclusively fine-tune the uppermost k layers, where $H \gg k$. Furthermore, we adopt a caching strategy wherein we precompute and store the hidden states from the lower layers for all contents within the dataset (potentially numbering in the thousands) prior to fine-tuning. For instance, in the case of the LLaMA-7B model comprising 32 layers, only the top 2 layers are subjected to fine-tuning. This caching process substantially mitigates computational costs, reducing the LLM's computation load to a mere $2/32 \approx 6\%$ of the original cost.

Parameter-Efficient Tuning. Low-Rank Adaptation (LoRA) [13] introduces trainable rank decomposition matrices into pretrained model layers, notably slashing the necessary trainable parameters for downstream tasks. This outperforms traditional fine-tuning, drastically reducing parameters, sometimes by a factor of 10,000. Here, we apply LoRA to the unfrozen Transformer layers, which are the most parameter-intensive components of the model. We also test finetuning without LoRA, employing *distinct learning rates* for the pretrained Transformer layers and other model components. Further elaboration is available in the Experiments section.

4 GENRE: PROMPTING CLOSED-SOURCE LLMS

Large language models differ significantly from previous models like BERT [9] in terms of their emergent abilities [39] such as strong text comprehension and language generation capabilities, having resulted in a paradigm shift from the traditional pretrain-finetune approach to the prompting-based approach. Previous studies [20] have found that using closed-source LLMs directly as recommenders without finetuning (completely bypassing conventional recommendation systems), using methods like prompts [24, 37] or in-context learning [6], only matches the performance of basic matrix factorization [16] methods or even random recommendations. This falls short when compared to modern attention-based approaches.

To overcome this, we propose a generative recommendation framework, namely **GENRE**, as shown in Figure 4a: leveraging closed-source LLMs (specifically, GPT-3.5) to augment data, aiming to enhance their performance on downstream conventional recommendation models. More precisely, the workflow 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** *(optional)*: use the generated data to update the current data for the next round of prompting and generation. **4) Training**: leverage the generated data to train news recommendation models. If the updating step is performed, we name it as "*Chain-based Generation*", otherwise, we name it as "*One-pass Generation*".

4.1 LLMs as Content Summarizer

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

By providing the content title, abstract, and category as input, the large language model produces a more informative title as output, as illustrated in Figure 4b. During downstream training, the enhanced content title will replace the original one and be used as one of the input features for the content encoder (Figure 3).

4.2 LLMs as User Profiler

The user profile generally refers to their preferences and characteristics, such as age, gender, topics of interest, and geographic location. They are usually not provided in the anonymized dataset due to privacy policies. Large language models are capable of understanding the browsing history and analyze an outline of the user profile.

As depicted in Figure 4b, the large language model produces topics and regions of interest when given the user browsing history. In this example, GPT-3.5 infers that the user may be interested in the region of "Florida", based on the word "Miami" in the news. While "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.

To incorporate the inferred user profile into the recommendation model, we first fuse the topics and regions of interest into an interest vector $\mathbf{v}_i$, defined by:

$$\mathbf{v}_i = \left[\text{POOL}\left(\mathbf{E}_{\text{topics}}\right); \text{POOL}\left(\mathbf{E}_{\text{regions}}\right)\right] \in \mathbb{R}^{2\times d}, \tag{5}$$

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. Then, the interest vector $\mathbf{v}_i$ will be combined with the user vector $\mathbf{v}_u$ learned from the history encoder (shown in Figure 3) to form the interest-aware user vector $\mathbf{v}_{iu}$ as follows:

$$\mathbf{v}_{iu} = \text{MLP}\left([\mathbf{v}_u; \mathbf{v}_i]\right) \in \mathbb{R}^d, \tag{6}$$

where MLP is a multi-layer perceptron with ReLU activation. Finally, the interest-aware user vector will replace the original user vector to participate in the click probability prediction.

4.3 LLMs as Personalized Content Generator

Recent studies [5, 32] have shown that large language models possess exceptional capabilities to learn from few examples. Hence, we propose to use GPT-3.5 to model the distribution of user-interested content given very limited browsing history data. Specifically, we use it as a personalized content generator to generate synthetic content that may be of interest to new users[6] have limited interaction data, making it difficult for the user encoder to capture their characteristics and ultimately weakening its ability to model warm users[7], enhancing their historical interactions and allowing the history encoder to learn effective user representations.

4.4 Chain-based Generation

While we have shown several examples of "one-pass generation", it is worth noting that large language models allow iterative generation and updating. The data generated by the large language models can be leveraged to enhance the quality of current data, which can subsequently be utilized in the next round of prompting and generation in an iterative fashion.

We design a chain-based personalized content generator by combining the one-pass user profiler and personalized content generator. Specifically, we first use the GPT-3.5 to generate the interested topics and regions of a user, which are then combined with the user history to prompt the large language model to generate synthetic content pieces. The user profile helps the large language models to engage in chain thinking, resulting in synthetic content that better matches the user's interests than the one-pass prompting.

[6]Following [12], we use "new users" to refer to users with no more than five contents in browsing history.

[7]We use "warm user" to represent the user who has browsed more than five contents.

Table 1: Data statistics. We use "user$_n$" to denote new users. Green numbers signify improvements over the original dataset, while blue numbers indicate the values of newly introduced features.

Dataset	MIND	Goodreads	Dataset	MIND	Goodreads
Original			*Content Summarizer (CS)*		
# content	65,238	16,833	**tokens/title**	+3.17	-
tokens/title	13.56	6.10	**tokens/desc**	-	29.28
# users	94,057	23,089	*User Profiler (UP)*		
# new user	20,110	2,306	**topics/user**	4.82	4.55
content/user	14.98	7.81	**regions/user**	0.29	-
content/user$_n$	3.19	3.03	*Personalized Content Generator (CG)*		
# pos	347,727	273,888	**#content**	+40,220	+4,612
# neg	8,236,715	485,233	**content/user$_n$**	+2.00	+2.00

5 EXPERIMENTS

5.1 Experimental Setup

Datasets. We conduct experiments on two real-world content-based recommendation dataset, i.e., news recommendation dataset MIND [45] and book recommendation dataset Goodreads [36]. In Table 1, we present the statistics of both the original dataset and the augmented versions. We use LLaMA-7B and LLaMA-13B models [30] as our open-source large language models, and GPT-3.5[8] as our closed-source model. For the augmented datasets, only the attributes that are different than the original datasets are shown in Table 1. For the Goodreads dataset, the content summarizer is used for the book description generation, given only the book title.

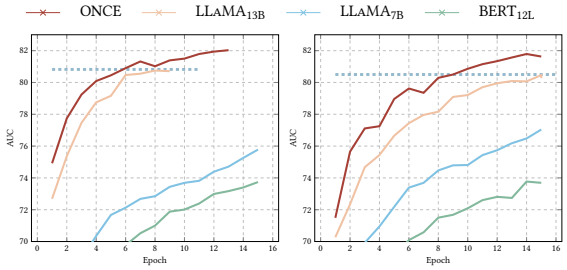
Recommendation Models. We evaluate the effectiveness of proposed ONCE method with three popular content-based recommendation models, namely NAML [40], NRMS [42], and Fastformer [44]. We also compare with PLM-NR [43] method, which replaces the original content encoder with small-scale pretrained language models such as BERT [9].

Evaluation Metrics. We follow the common practice [25, 42, 43] to evaluate the effectiveness of news recommendation models with the widely used metrics, i.e., AUC [10], MRR [35] and nDCG [14]. In this work, we use nDCG@1 and nDCG@5 for evaluation on the Goodreads dataset, and nDCG@5 and nDCG@10 for evaluation on the MIND dataset shortly denoted as N@1, N@5 and N@10, respectively.

[8] https://platform.openai.com/docs/guides/chat

Table 2: Performance comparison among original recommenders, recommenders enhanced by open-source LLMs (i.e., DIRE), those enhanced by closed-source LLMs (i.e., GENRE), and those boosted by both types of LLMs (i.e., ONCE). Within the closed-source LLM category, the abbreviations "CS", "UP", "CG", and "UP→CG" represent datasets augmented by the one-pass content summarizer, one-pass user profiler, one-pass personalized content generator, and chain-based personalized content generator, respectively. Furthermore, "ALL" denotes a dataset that incorporates enhancements from CS, and UP→CG. The top-performing results are emphasized in bold.

		NAML (2019a)				NRMS (2019c)				Fastformer (2021b)				MINER (2022)			
		AUC	MRR	N@5	N@10	AUC	MRR	N@5	N@10	AUC	MRR	N@5	N@10	AUC	MRR	N@5	N@10
MIND dataset																	
	Original	61.75	30.60	31.35	37.85	61.71	30.20	30.98	37.42	62.26	31.14	31.90	38.32	63.88	32.19	33.04	39.45
DIRE	**BERT$_{12L}$** [43]	65.32	33.16	34.29	40.35	64.08	31.24	32.35	38.66	65.48	32.47	33.41	39.75	65.82	32.77	34.02	40.19
	LLaMA$_{7B}$ *(Ours)*	68.34	35.80	37.60	43.48	68.50	36.21	38.11	43.91	68.55	36.59	38.38	44.06	68.70	36.58	38.49	44.18
	LLaMA$_{13B}$ *(Ours)*	68.23	35.99	37.93	43.77	68.45	36.15	38.02	43.88	68.51	36.37	38.20	44.02	68.59	36.46	38.38	44.05
GENRE	**CS** *(Ours)*	63.73	31.83	32.94	39.24	63.85	31.57	32.35	38.80	64.73	32.81	33.68	40.06	65.71	33.59	34.90	40.96
	UP *(Ours)*	62.19	30.90	31.78	38.26	61.90	30.60	31.54	37.66	63.40	31.94	32.76	39.15	64.45	32.09	33.14	39.54
	CG *(Ours)*	62.93	30.83	32.10	38.34	63.04	31.00	31.84	38.22	64.69	32.28	33.31	39.76	64.21	32.30	33.57	39.91
	UP→CG *(Ours)*	63.61	31.58	32.63	39.07	62.95	32.00	32.80	39.00	64.82	32.44	33.51	39.93	64.73	33.09	34.10	40.32
	ALL *(Ours)*	63.88	32.17	33.14	39.37	63.71	32.14	33.11	39.43	66.70	34.20	35.81	41.78	66.46	34.20	35.47	41.48
	ONCE *(ours)*	**68.62**	**36.50**	**38.31**	**44.05**	**68.74**	**36.66**	**38.60**	**44.37**	**68.83**	**36.68**	**38.56**	**44.35**	**68.92**	**36.74**	**38.72**	**44.48**
	Improvement (%) over Original	**11.13%**	**19.28%**	**22.20%**	**16.38%**	**11.39%**	**21.39%**	**24.60%**	**18.57%**	**10.55%**	**17.79%**	**20.88%**	**15.74%**	**7.89%**	**14.13%**	**17.19%**	**12.75%**
	Improvement (%) over BERT$_{12L}$	**5.05%**	**10.07%**	**11.72%**	**9.17%**	**7.27%**	**17.35%**	**19.32%**	**14.77%**	**5.12%**	**12.97%**	**15.41%**	**11.57%**	**4.71%**	**12.11%**	**13.82%**	**10.67%**
Goodreads dataset																	
	Original	66.47	75.75	58.49	82.20	68.95	77.05	60.62	83.16	70.85	78.37	62.90	84.15	71.03	78.46	63.09	84.20
DIRE	**BERT$_{12L}$** [43]	70.68	78.17	62.26	83.99	71.80	78.87	63.62	84.51	72.47	79.29	64.45	84.82	73.36	80.08	65.19	85.25
	LLaMA$_{7B}$ *(Ours)*	77.01	82.74	71.09	89.39	75.90	81.75	69.13	86.65	76.52	82.31	70.48	87.03	76.45	82.46	70.31	86.92
	LLaMA$_{13B}$ *(Ours)*	77.43	83.05	71.56	87.61	77.57	82.96	71.41	87.55	77.46	83.00	71.36	87.58	77.50	83.07	71.44	87.64
GENRE	**CS** *(Ours)*	67.68	76.41	59.64	82.69	69.77	77.57	61.54	83.35	71.41	78.77	63.70	84.43	71.96	79.09	64.30	84.72
	UP *(Ours)*	68.45	76.91	60.70	83.08	69.45	77.58	61.89	83.57	71.15	78.68	63.86	84.39	71.67	78.85	63.94	84.50
	CG *(Ours)*	66.94	76.10	59.26	82.47	70.09	77.95	62.34	83.83	71.08	78.53	63.38	84.26	71.81	78.89	63.99	84.53
	UP→CG *(Ours)*	67.98	76.78	60.56	82.96	69.95	77.79	62.07	83.71	71.88	79.02	64.10	84.63	71.79	78.93	63.97	84.56
	ALL *(Ours)*	68.95	77.25	61.19	83.32	72.07	79.13	64.46	84.72	73.23	79.97	66.07	85.33	73.21	79.91	65.73	85.29
	ONCE *(ours)*	**77.63**	**83.13**	**71.65**	**87.66**	**77.89**	**83.31**	**71.89**	**87.79**	**78.03**	**83.52**	**72.52**	**87.96**	**77.82**	**83.35**	**71.96**	**87.85**
	Improvement (%) over Original	**16.79%**	**9.74%**	**22.50%**	**6.64%**	**12.97%**	**8.12%**	**18.59%**	**5.57%**	**10.13%**	**6.57%**	**15.29%**	**4.53%**	**9.56%**	**6.23%**	**14.06%**	**4.33%**
	Improvement (%) over BERT$_{12L}$	**9.83%**	**6.35%**	**15.08%**	**4.37%**	**8.48%**	**5.63%**	**13.00%**	**3.88%**	**7.67%**	**5.33%**	**12.52%**	**3.70%**	**6.08%**	**4.08%**	**10.39%**	**3.05%**



(a) NRMS **(b) Fastformer**

Figure 5: Training curves for open-source LLMs and ONCE. The y-axis AUC value is evaulated on the validation set.

Implementation Details. During training, we employ Adam [15] optimizer with a learning rate of 1e-3 for the MIND dataset and 1e-4 for the Goodreads dataset. If the large language models are not tuned with LoRA [13], their learning rates are set to 1e-5. For all models, the embedding dimension of non-LLM modules is set to 64, and the negative sampling ratio is set to 4. We tune the hyperparameters of all base models to attain optimal performance. We average the results of five independent runs for each model and observe the p-value smaller than 0.01. All LLaMA-based experiments are conducted on a single NVIDIA A100 device with 80GB memory, and others on a single NVIDIA GeForce RTX 3090 device. We release all our code and datasets[9] for other researches to reproduce our work.

5.2 Performance Comparison

Table 2 provides an overview of the performance enhancements observed across four base models on two datasets, boosted by open-source LLM, closed-source LLM, and dual LLM (i.e., ONCE) approaches. Drawing from the results, we can derive the following observations:

Firstly, the open-source LLM group exhibits substantial improvements in the base models. The pretraining of LLaMA endows it with robust semantic understanding and a wealth of content-level knowledge, including elements like book titles and geographic locations. Additionally, its high-dimensional representation space ensures efficient encoding of extensive information within hidden

[9] https://github.com/Jyonn/ONCE

Table 3: Influence of the number of frozen layers on three open-source LLMs. Best results are highlighted in bold, while results inferior to the respective base models are indicated in red. "F/T" denotes the number of frozen and tuning layers, respectively.

		NAML (2019a)				**NRMS** (2019c)				**Fastformer** (2021b)				**MINER** (2022)			
Encoder	**F/T**	**AUC**	**MRR**	**N@5**	**N@10**	**AUC**	**MRR**	**N@5**	**N@10**	**AUC**	**MRR**	**N@5**	**N@10**	**AUC**	**MRR**	**N@5**	**N@10**
MIND dataset																	
Original	-	61.75	30.60	31.35	37.85	61.71	30.20	30.98	37.42	62.26	31.14	31.90	38.32	63.88	32.19	33.04	39.45
BERT$_{12L}$	12/0	65.32	33.16	34.29	40.35	64.08	31.24	32.35	38.66	64.25	32.05	32.88	39.17	64.75	32.44	33.60	39.87
	11/1	65.10	32.86	33.99	40.19	62.59	31.46	32.09	38.61	65.48	32.47	33.41	39.75	65.82	32.77	34.02	40.19
	10/2	63.79	32.27	32.95	39.40	62.68	30.95	31.61	37.89	63.41	31.57	32.56	38.92	64.01	31.69	32.82	39.17
LLaMA$_{7B}$	32/0	67.78	35.17	36.84	42.78	68.10	35.33	36.91	43.04	67.83	35.19	36.57	42.59	67.96	35.28	36.72	42.80
	31/1	**68.34**	35.80	37.60	43.48	68.33	35.81	37.43	43.37	68.51	36.56	**38.46**	**44.15**	68.45	36.41	38.25	43.93
	30/2	68.18	**36.09**	37.76	43.65	**68.50**	**36.21**	**38.11**	**43.91**	68.55	**36.59**	38.38	44.06	**68.70**	**36.58**	**38.49**	**44.18**
LLaMA$_{13B}$	40/0	68.23	35.99	**37.93**	**43.77**	68.45	36.15	38.02	43.88	68.51	36.37	38.20	44.02	68.59	36.46	38.38	44.05
	39/1	67.66	35.73	37.59	43.35	68.23	36.05	37.97	43.72	**68.60**	36.45	38.27	43.96	68.53	36.37	38.21	44.00
	38/2	68.19	36.07	37.89	43.68	68.30	36.13	37.95	43.74	68.19	35.96	37.72	43.50	67.83	35.88	37.64	43.45
Goodreads dataset																	
Original	-	66.47	75.75	58.49	82.20	68.95	77.05	60.62	83.16	70.85	78.37	62.90	84.15	71.03	78.46	63.09	84.20
BERT$_{12L}$	12/0	62.05	72.82	53.37	80.06	64.49	74.38	56.47	81.22	66.83	75.85	58.68	82.35	67.11	76.09	58.88	82.48
	11/1	62.32	73.03	53.82	80.22	65.94	75.35	58.12	81.94	66.23	75.53	58.12	82.05	66.72	75.93	58.60	82.28
	10/2	65.22	74.90	57.07	81.58	63.77	73.94	55.53	80.88	67.66	76.49	60.03	82.76	67.94	76.72	60.22	82.89
	0/12	70.68	78.17	62.26	83.99	71.80	78.87	63.62	84.51	72.47	79.29	64.45	84.82	73.36	80.08	65.19	85.25
LLaMA$_{7B}$	32/0	69.29	77.32	60.89	83.37	71.96	79.19	64.73	84.77	72.25	79.16	64.35	84.73	71.14	78.53	63.44	84.27
	31/1	73.82	80.34	66.51	85.61	75.18	81.23	68.04	86.27	75.80	81.70	68.69	86.58	75.33	81.35	68.26	86.33
	30/2	77.01	82.74	71.09	**89.39**	75.90	81.75	69.13	86.65	76.52	82.31	70.48	87.03	76.45	82.46	70.31	86.92
LLaMA$_{13B}$	40/0	70.31	78.00	62.36	83.88	72.82	79.79	65.79	85.21	71.66	78.81	63.52	84.48	73.28	80.13	66.09	85.23
	39/1	**77.43**	**83.05**	**71.56**	87.61	76.55	82.32	70.08	87.07	76.42	82.36	70.51	87.10	77.18	82.60	71.17	87.43
	38/2	76.25	82.18	69.98	86.98	**77.57**	**82.96**	**71.41**	**87.55**	**77.46**	**83.00**	**71.36**	**87.58**	**77.50**	**83.07**	**71.44**	**87.64**

states. **Secondly**, the closed-source LLM group also demonstrates impressive performance, highlighting the efficacy of data enrichment in introducing enhanced semantic features to the dataset. The fusion of diverse prompt techniques (i.e., "ALL") further amplifies model effectiveness. **Thirdly**, our dual LLM-based ONCE method showcases additional performance gains compared to employing a single LLM, albeit the improvement is relatively modest compared to the open-source finetuning. GPT-3.5 offers LLaMA with supplementary semantic insights, elevating its content comprehension capabilities. However, the closed-source LLM contributes token-level discrete features, which bear less influence when juxtaposed with the continuous embedding-level representations delivered by open-source LLMs.

In addition,Figure 5 presents the training curves for open-source LLMs and ONCE. Notably, ONCE (using LLaMA-13B as the backbone), leveraging closed-source LLM information, demonstrates both a stronger initial performance and quicker training efficiency. Specifically, on the NRMS model, ONCE reaches performance equivalent to LLaMA-13B's 8th epoch by its 6th epoch, a substantial 25% improvement. On the Fastformer model, ONCE surpasses LLaMA-13B's 15th epoch performance by its 9th epoch, showcasing an impressive 40% enhancement.

5.3 Ablation Study on Open-source LLMs

Here, we study the impact of the finetuning layers and low-rank adaptation (LoRA) on the performance of open-source LLMs.

Table 3 presents a comparison of finetuning effects on the top $0 \sim 2$ layers of transformers across different open-source LLMs. Key findings from the results include:

Firstly, In most instances, substantial enhancements in recommendation models are evident even without finetuning (T=0) the LLMs. Notably, the BERT model on the Goodreads dataset is an exception due to the unique challenge posed by book titles as content, which lacks the enriched knowledge of LLaMA, resulting in less effective representations primarily focused on literal meanings. **Secondly**, within the MIND dataset, LLaMA-7B generally outperforms LLaMA-13B with finetuning $1 \sim 2$ layers. This might stem from the relative difficulty in fine-tuning LLaMA-13B, while the 7B model sufficiently captures the semantic richness of news headlines. Conversely, for the Goodreads dataset, LLaMA-13B demonstrates the most promising outcomes. **Thirdly**, overall, a greater number of tuned layers correlates with improved performance, though this also entails increased training costs.

Table 4 presents the influence of LoRA during the finetuning process of open-source LLMs. Our findings indicate that, for the

Table 4: Influence of the use of low-rank adaption (LoRA). The experiments are conducted over the NAML model.

	LoRa				w/o LoRa			
Encoder	**AUC**	**MRR**	**N@5**	**N@10**	**AUC**	**MRR**	**N@5**	**N@10**
MIND dataset								
$\textbf{BERT}_{12l}$	65.10	32.86	33.99	40.19	62.94	31.32	32.20	38.52
$\textbf{LLaMA}_{7B}$	68.34	35.80	37.60	43.48	67.25	34.28	36.00	42.12
Goodreads dataset								
$\textbf{BERT}_{12L}$	63.18	76.80	55.37	80.69	70.68	78.17	62.26	83.99
$\textbf{LLaMA}_{7B}$	75.00	81.23	68.44	86.29	77.01	82.74	71.09	89.39

Table 5: Effectiveness of the personalized content generator (CG) for both new user and warm user groups, assessed on the MIND dataset. ORI: training with the original data. Imp.: denotes the improvement realized through the personalized content generator.

		New User				Warm User			
		AUC	**MRR**	**N@5**	**N@10**	**AUC**	**MRR**	**N@5**	**N@10**
NAML	**ORI**	59.24	**32.82**	34.24	**40.34**	62.21	30.20	30.83	37.40
	CG	**60.21**	32.69	**34.67**	40.33	**63.43**	**30.49**	**31.64**	**37.98**
	Imp.	0.97	-	0.43	-	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
	CG	**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

MIND dataset, LoRA leads to improved performance, while a different pattern emerges for the Goodreads dataset. This divergence might be attributed to differences in the nature of the input textual data. Goodreads employs book titles with relatively limited informative content, whereas MIND's input news headlines inherently encapsulate the core essence of the content. Constructing a robust representation from book titles requires more nuanced adjustments of the network parameters.

5.4 Ablation Study on Closed-source LLMs

Here, we investigate the impact of the synthetic content data on two user groups, i.e., new user group and warm user group. From the results in Table 5, it can be seen that the personalized content generator improves the performance of both the new and warm user groups in most cases. This is because the history encoder struggles to capture the interests of new users due to their limited history, which also affects its ability to model warm users. With the generated content pieces added to the history of new users, the history encoder can better capture their interests, leading to a performance improvement on both groups.

6 RELATED WORKS

6.1 LLMs for Recommendation

The recent advancement of Large Language Models (LLMs) like ChatGPT and LLaMa [30], has triggered a new wave of interest, resulting in the development of diverse applications across multiple domains [5, 27, 47]. Using self-supervised learning on large datasets, these models excel in text representation and, with transfer techniques such as fine-tuning and prompt tuning, they hold the potential to enhance recommendation systems, gaining notable attention in the RS domain.

According to the categorization proposed by Lin *et al.* [21], the application of LLMs in recommendation systems can be segmented into five categories based on their position in the pipeline: User data collection, Feature engineering (e.g., [3]), Feature encoder (e.g., [50]), Scoring/Ranking function (e.g., [18, 22]), and Pipeline controller (e.g., [39]); alternatively, they can also be grouped into four types, considering two dimensions: (1) whether they are a tune LLM and (2) whether they infer in conjunction with conventional recommendation models (CRMs). In our study, we employed LLMs for dataset enhancement (feature engineering) and encoding content features, which were subsequently integrated into CRMs. To the best of our knowledge, we are the first to combine the open- and closed-source LLMs in recommendation.

6.2 Content-based Recommendation

Content-based recommendations encompass a diverse range of domains, including but not limited to music [4, 33, 38], news [11, 23], and videos [7, 8, 17]. In this paper, our primary focus is on the news and book recommendation.

To better capture textual knowledge and user preferences in news recommendation, in the past few years, several models based on deep neural networks have been proposed [1, 40–42]. Despite their effectiveness, these end-to-end models have limited semantic comprehension abilities. In recent years, there has been a surge of interest in using pretrained language models (PLMs) such as BERT [9] and GPT [28] in news recommendation systems [25, 43, 49], owing to the powerful transformer-based architectures and the availability of large-scale pretraining data. The emergence of LLMs has further offered potential to enhance recommender systems using its rich general knowledge [46]. In the latest developments, LLMs have been applied to personalization [29] and product recommendation [18]. Nevertheless, [24] points out that directly employing LLMs as a recommender system has shown negative results, indicating that the use of LLMs for news recommendation remains understudied.

7 CONCLUSION

Our work addresses the limitations of content-based recommendation systems and offers a new approach that leverages both open- and closed-source LLMs to enhance their performance. Our findings indicate that combining the finetuning on the open-source LLMs and the prompting on the closed-source LLMs into recommendation systems can lead to substantial improvements, which has important implications for online content platforms. Our ONCE framework can be applied to other content-based domains beyond news and book recommendation. We hope our work will encourage further research and contribute to the development of more effective recommendation systems based on large language models.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable feedback.

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] Vadim Borisov, Kathrin Sessler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. 2022. Language Models are Realistic Tabular Data Generators. In *The Eleventh International Conference on Learning Representations*.

[4] Jiajun Bu, Shulong Tan, Chun Chen, Can Wang, Hao Wu, Lijun Zhang, and Xiaofei He. 2010. Music Recommendation by Unified Hypergraph: Combining Social Media Information and Music Content *(MM '10)*. Association for Computing Machinery, New York, NY, USA, 391–400. https://doi.org/10.1145/1873951.1874005

[5] 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]

[6] 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]

[7] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, and Dasarathi Sampath. 2010. The YouTube Video Recommendation System. In *Proceedings of the Fourth ACM Conference on Recommender Systems* (Barcelona, Spain) *(RecSys '10)*. Association for Computing Machinery, New York, NY, USA, 293–296. https://doi.org/10.1145/1864708.1864770

[8] Yashar Deldjoo, Mehdi Elahi, Paolo Cremonesi, Franca Garzotto, Pietro Piazzolla, and Massimo Quadrana. 2016. Content-based video recommendation system based on stylistic visual features. *Journal on Data Semantics* 5 (2016), 99–113.

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

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

[11] Florent Garcin, Christos Dimitrakakis, and Boi Faltings. 2013. Personalized News Recommendation with Context Trees. In *Proceedings of the 7th ACM Conference on Recommender Systems* (Hong Kong, China) *(RecSys '13)*. Association for Computing Machinery, New York, NY, USA, 105–112. https://doi.org/10.1145/2507157.2507166

[12] 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.

[13] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).

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

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

[16] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[17] Joonseok Lee and Sami Abu-El-Haija. 2017. Large-Scale Content-Only Video Recommendation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*.

[18] 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]

[19] Jian Li, Jieming Zhu, Qiwei Bi, Guohao Cai, Lifeng Shang, Zhenhua Dong, Xin Jiang, and Qun Liu. 2022. MINER: Multi-Interest Matching Network for News Recommendation. In *Findings of the Association for Computational Linguistics: ACL 2022*. 343–352.

[20] Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Xiangyang Li, Chenxu Zhu, Huifeng Guo, Yong Yu, Ruiming Tang, et al. 2023. How Can Recommender Systems Benefit from Large Language Models: A Survey. *arXiv preprint arXiv:2306.05817* (2023).

[21] Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Xiangyang Li, Chenxu Zhu, Huifeng Guo, Yong Yu, Ruiming Tang, and Weinan Zhang. 2023. How Can Recommender Systems Benefit from Large Language Models: A Survey. arXiv:2306.05817 [cs.IR]

[22] Guang Liu, Jie Yang, and Ledell Wu. 2022. PTab: Using the Pre-trained Language Model for Modeling Tabular Data. arXiv:2209.08060 [cs.LG]

[23] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized News Recommendation Based on Click Behavior. In *Proceedings of the 15th International Conference on Intelligent User Interfaces* (Hong Kong, China) *(IUI '10)*. Association for Computing Machinery, New York, NY, USA, 31–40. https://doi.org/10.1145/1719970.1719976

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

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

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

[30] 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).

[31] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

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

[33] Aaron van den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. In *Advances in Neural Information Processing Systems*, C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger (Eds.), Vol. 26. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2013/file/b3ba8f1bee1238a2f37603d90b58898d-Paper.pdf

[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] Mengting Wan and Julian McAuley. 2018. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM conference on recommender systems*. 86–94.

[37] Lei Wang and Ee-Peng Lim. 2023. Zero-Shot Next-Item Recommendation using Large Pretrained Language Models. *arXiv preprint arXiv:2304.03153* (2023).

[38] Xinxi Wang and Ye Wang. 2014. Improving Content-Based and Hybrid Music Recommendation Using Deep Learning. In *Proceedings of the 22nd ACM International Conference on Multimedia* (Orlando, Florida, USA) *(MM '14)*. Association for Computing Machinery, New York, NY, USA, 627–636. https://doi.org/10.1145/2647868.2654940

[39] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *Trans. Mach. Learn. Res.* (2022).

[40] 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*.

[41] 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

[42] 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.

[43] 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.

[44] Chuhan Wu, Fangzhao Wu, Tao Qi, Yongfeng Huang, and Xing Xie. 2021. Fastformer: Additive attention can be all you need. *arXiv preprint arXiv:2108.09084* (2021).

[45] 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.

[46] Jiahao Wu, Qijiong Liu, Hengchang Hu, Wenqi Fan, Shengcai Liu, Qing Li, Xiao-Ming Wu, and Ke Tang. 2023. Leveraging Large Language Models (LLMs) to Empower Training-Free Dataset Condensation for Content-Based Recommendation. *arXiv preprint arXiv:2310.09874* (2023).

[47] 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]

[48] Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. 2023. Where to go next for recommender systems? id-vs. modality-based recommender models revisited. *arXiv preprint arXiv:2303.13835* (2023).

[49] 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*.

[50] Qi Zhang, Jingjie Li, Qinglin Jia, Chuyuan Wang, Jieming Zhu, Zhaowei Wang, and Xiuqiang He. 2021. UNBERT: User-News Matching BERT for News Recommendation. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Zhi-Hua Zhou (Ed.). International Joint Conferences on Artificial Intelligence Organization, 3356–3362. https://doi.org/10.24963/ijcai.2021/462 Main Track.