Leveraging ChatGPT to Empower Training-free Dataset Condensation for Content based Recommendation

Jiahao Wu*
PolyU & SUSTech
Hong Kong SAR & Shenzhen, China
jiahao.wu@connect.polyu.hk

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

Hengchang Hu
National University of Singapore
Singapore, Singapore
holdenhhc@gmail.com

Wenqi Fan†
The Hong Kong Polytechnic University
Hong Kong SAR, China
wenqi.fan@polyu.edu.hk

Shengcai Liu†
Southern University of Science and Technology
Shenzhen, China
liusc3@sustech.edu.cn

Qing Li
The Hong Kong Polytechnic University
Hong Kong SAR, China
csqli@comp.polyu.edu.hk

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

Ke Tang
Southern University of Science and Technology
Shenzhen, China
tangk3@sustech.edu.cn

Abstract

Modern Content-Based Recommendation (CBR) techniques utilize item content to deliver personalized services, effectively mitigating information overload. However, these methods often require resource-intensive training on large datasets. To address this issue, we explore dataset condensation for textual CBR in this paper. Dataset condensation aims to synthesize a compact yet informative dataset, enabling models to achieve performance comparable to those trained on full datasets. Applying existing approaches to CBR presents two key challenges: (1) the difficulty of synthesizing discrete texts and (2) the inability to preserve user-item preference information. To overcome these limitations, we propose TF-DCon, an efficient dataset condensation method for CBR. TF-DCon employs a prompt-evolution module to guide ChatGPT in condensing discrete texts and integrates a clustering-based module to condense user preferences effectively. Extensive experiments conducted on three real-world datasets demonstrate TF-DCon's effectiveness. Notably, we are able to approximate up to 97% of the original performance while reducing the dataset size by 95% (i.e., dataset MIND). We have released our code and data here for other researchers to reproduce our results[1].

CCS Concepts

• **Information systems** → **Recommender systems**; **Summarization**; **Language models**.

Keywords

Recommender system, Dataset condensation, Large language model

ACM Reference Format:
Jiahao Wu, Qijiong Liu, Hengchang Hu, Wenqi Fan, Shengcai Liu, Qing Li, Xiao-Ming Wu, and Ke Tang. 2025. Leveraging ChatGPT to Empower Training-free Dataset Condensation for Content based Recommendation. In *Companion Proceedings of the ACM Web Conference 2025 (WWW Companion '25), April 28-May 2, 2025, Sydney, NSW, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3701716.3715555

1 Introduction

Content-based recommenders [22, 24, 25] have made strides in mitigating the information overload dilemma, delivering items with relevant content (i.e., news, articles, or movies) to users. Existing advanced content-based recommendation (CBR) models [20, 23] are trained on large-scale datasets encompassing millions of users and items. Training on these large datasets heavily strains computational resources. Furthermore, the recurrent retraining of recommendation models, especially for periodic updates in real-world applications, exponentially elevates costs to unsustainable levels.

Dataset condensation, also called *dataset distillation*, offers promising solution to these issues. The goal of dataset condensation is to synthesize a small yet informative dataset, trained on which the model can achieve comparable performance to that of a model trained on the original dataset [10, 12, 13, 21, 28, 29]. The conventional paradigm [12, 17, 28] formulates the condensation as bi-level optimization problem and iteratively update the condensed data by matching gradients of network parameters between synthetic

*Both authors contributed equally to this research.
†Corresponding authors.
[1]https://github.com/Jyonn/TF-DCon

WWW Companion '25, April 28-May 2, 2025, Sydney, NSW, Australia

ACM ISBN 979-8-4007-1331-6/25/04
https://doi.org/10.1145/3701716.3715555

and original data. Such methods have achieved success on condensing continuous data, such as images and node features. However, condensing datasets for CBR is still unexplored.

To bridge this gap, we investigate how to effectively condense the dataset for the textual content based recommendation. Existing condensation approaches devised in other domains [10, 12, 13, 28] face two main challenges in the context of CBR: (1) *How to generate discrete textual data?* Current condensation methods are designed for continuous data (i.e., images and text embeddings), following a formulation of nested bi-level optimization. Under this formulation, the data is synthesized via the outer gradient, which cannot be utilized to generate discrete textual data. Therefore, a solution for text synthesis is essential. (2) *How to preserve the preference information between users and items?* In recommendation tasks, user and item data, along with their interactions, are critical for inferring user preferences. However, previously proposed methods mostly involve classification tasks, limiting their ability to handle interaction data and retain preference information.

Large Language Models (LLMs), such as ChatGPT developed by OpenAI, have shown a strong capacity to distill textual information [3, 5], and its general expertise across various fields such as medicine [1, 2], recommendation [8, 9, 15, 16], and law [4, 6]. With their versatility and extensive world knowledge, LLMs exhibit emergent abilities such as davanced text comprehension and language generation [3, 5]. Leveraging these capabilities [3, 5], we propose to leverage ChatGPT for dataset condensation, processing textual content to address one of the aforementioned challenges.

While LLMs are highly effective at processing text, they lack specific domain knowledge for recommendation scenarios and the ability to capture personalized user preferences. Naïvely applying LLMs for recommendation data condensation risks losing both this domain knowledge and essential preference information.

To this end, we propose a ChatGPT-powered **T**raining-**F**ree **D**ataset **Con**densation method for content-based recommendation, abbreviated as **TF-DCon**. TF-DCon is devised in a two-level manner: *content-level* and *user-level*. At content-level, we curate a prompt-evolution module to optimize prompts, enabling ChatGPT to adapt to the specific recommendation domain and condense each item's information into an informative title. At user-level, to capture the preference information of users, we propose a clustering-based synthesis module to simultaneously generate fake users and their corresponding historical interactions, based on user interests extracted by ChatGPT and user embeddings. Consequently, our approach offers several advantages: (1) TF-DCon can generate text, which makes the condensed dataset more generalizable and flexible to train different architectures of models for CBR. (2) TF-DCon seamlessly embeds user preferences into the condensed data through the clustering-based synthesis. (3) TF-DCon formulates condensation as a forward process, eliminating iterative training in the bilevel paradigm [13, 26, 28].

2 Preliminaries

Content-based Recommendation. CBR aims to infer the preference of users based on historical interactions and then recommend the contents with potential interest. The dataset $\mathcal{D}$ mainly consists of item set $\mathcal{N}$, user set $\mathcal{U}$, and click history set $\mathcal{H}$. Each item $n \in \mathcal{N}$ contains various contents, such as title, abstract, and category. Each user $u \in \mathcal{U}$ has a clicking history of items $h^{(u)} \subset \mathcal{N}$. We denote the set of history $h^{(u)}$ by $\mathcal{H}$. Let C denote the set of clicks, where each click $c \in C$ is defined as a tuple (u, n) indicating that user u has clicked on item n. CBR predicts user preferences for candidate items based on these interactions.

Dataset Condensation. Dataset condensation aims to synthesize a small yet informative dataset $\mathcal{S}$, trained on which the model can achieve comparable performance to the model trained on full dataset $\mathcal{D}$. Here, the synthesized dataset $\mathcal{S}$ also consists of its corresponding item set $\mathcal{N}^{\mathcal{S}}$, user set $\mathcal{U}^{\mathcal{S}}$ and click history set $\mathcal{H}^{\mathcal{S}}$. Formally, the objective of the condensation is as follows:

$$\min_{\mathcal{S}} \mathcal{L}(f_{\theta^{\mathcal{S}}}(\mathcal{D})), \text{ s.t. } \theta^{\mathcal{S}} = \arg\min_{\theta} \mathcal{L}(f_{\theta}(\mathcal{S})), \tag{1}$$

where θ is the parameter of recommendation model f and $\mathcal{L}$ is the loss function. In Equation 1, the condensation is a bi-level problem, where the outer optimization is to synthesize the condensed dataset and the inner optimization is to train the model on dataset $\mathcal{S}$.

3 Method

In this section, we detail the proposed method (TF-DCon) and the overview is shown in Figure 1. Content-level condensation reduces the textual data load for each item (Section 3.1), while user-level condensation synthesizes fake users and interactions (Section 3.2).

3.1 Content-level Condensation

Content Condensation. In the content-level condensation, we aim to condense all the information of each item into a succinct yet informative title. Recent studies [7, 18] reveal the exceptional power of large language models in processing textual contents. Therefore, we propose to utilize ChatGPT to condense the contents. Specifically, we design the guiding prompt in the following format:

> **Hints on the format of input:** [*title*]{*title*}, [*abs*]{*abs*}, ...
> **Hints on the format of output:** [*new_title*]{*new_title*}

After facilitating ChatGPT's comprehension on the input, the information of items will be fed into ChatGPT to generate the condensed title. Formally, the process can be described as follows:

$$n_s = ChatGPT(n), \tag{2}$$

where $n \in \mathcal{N}$ is the original contents of item, consisting of $[title]$, $[abstract]$ and $[category]$, and $n_s \in \mathcal{N}^{\mathcal{S}}$ is the condensed title of the item, which only contain the $[title]$.

Prompt Evolution. To efficiently guide ChatGPT to adapt to the recommendation scenario and perform thorough content condensation for each item, we propose a curated prompt evolution method. Given an initial prompt, EvoPro evolves the prompts over a pre-defined times. During i-th iteration of evolving, EvoPro first instructs ChatGPT to generate N next-generation prompts $\{prompt_n\}$ based on gen_{i-1}-*prompt*. Then, these prompts are utilized to instruct ChatGPT to condense item contents for TF-DCon. The similarity scores between the embeddings of condensed contents and the original contents will be assigned to each prompt in $\{prompt_n\}$. The highest-scoring prompt is selected as gen_i-*prompt* for i-th generation. When the evolution finished, the prompt with the hightest score in the latest generation will be selected as the

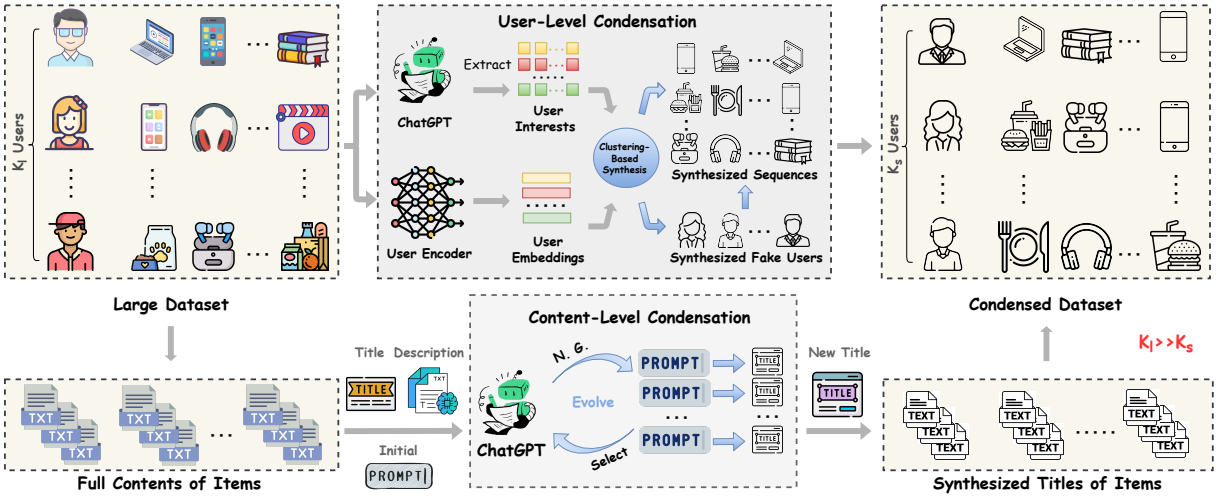


Figure 1: The Proposed Method in a Nutshell and "N. G." denotes "Next Generated Prompt Candidates".

prompt for content condensation. The prompt selection is based on the rationale that a condensed title with the highest similarity to the original content retains the most information.

3.2 User-level Condensation

In user-level condensation, we aim to condense the number of users and interaction size. Here, we reduce users and interactions by clustering user embeddings to generate synthetic users and their histories. Each cluster represents a synthetic user, and interactions are merged from top-m closest users to the cluster centroid.

Interests Extraction and User Encoder.

Interests Extraction: We utilize ChatGPT to extract the set of interests $\mathcal{I}^{(u)}$ for user u from his click history $h^{(u)}$. Specifically, we design the guiding prompt in the following format:

> **Hints on the format of input:** (1){*title*}, (2){*title*}, (3){*title*}, ...
> **Hints on the format of output:** [*interests*] -*inter1*, -*inter2*, ...

Then, the click history $h^{(u)}$ will be fed into ChatGPT to generate the interests of user u, which could be formulated as follows:

$$\mathcal{I}^{(u)} = ChatGPT(h^{(u)}), \tag{3}$$

where $\mathcal{I}^{(u)}$ is the set of interests for user u.

User Encoder: For each user u, given the associated click history $h^{(u)}$, we first we obtain the user's embedding:

$$z_u = f_\theta(h^{(u)}, \mathcal{N}), \tag{4}$$

where $\mathcal{N}$ is item set and f_θ is user encoder [22–24, 27].

Clustering-based Synthesis.

User Synthesis: Given the users' embeddings, we apply K-means algorithm over them to obtain their cluster centroids $\{c_i\}_{i=1}^{K_s}$, where K_s is the number of synthesized users in the condensed dataset.

Historical Sequence Synthesis: Given that each cluster corresponds to a fake user u_s, we need to synthesize the corresponding historical interactions. We devise a scoring module using user interests and user embeddings to select the historical interactions for fake users.

First, we calculate the distance between the embedding of each real user u and its corresponding prototype c_i as follows:

$$d_u^{emb} = Dis(z_u, c_i), \tag{5}$$

where $Dis(\cdot,\cdot)$ is the distance function.

Given a set of interests $\mathcal{I}^{(u)}$ for each user u, we encode those interests by pretrained language model (PLM) as follows:

$$e_u = f_{pool}\left(f_{PLM}(\mathcal{I}^{(u)})\right), \tag{6}$$

where f_{pool} is the pooling function and $f_{PLM}(\cdot)$ is the pretrained language model. Based on the clusters of user embeddings, we calculate the corresponding cluster centroids of user interests $\{c_i'\}_{i=1}^{K_s}$ by averaging the user interests in each cluster. Given the interest embeddings and their centroids, we can calculate the distance between them as follows:

$$d_u^{int} = Dis(e_u, c_i'). \tag{7}$$

Combining Equation 5 and Equation 7, we have:

$$d_u = d_u^{emb} + \alpha \cdot d_u^{int}, \tag{8}$$

where α is a hyperparameter and d_u is defined as the *selection score*. Within each cluster, we employ an ascending ordering of users based on their selection scores. We subsequently curate the historical interactions for a synthetic user by merging the interaction histories of the top-m users, which could be formalized as follows:

$$h_s^{(u_s)} = \cup_{i=1}^{m} h^{(u_i)}, \tag{9}$$

where d_{u_i} ranks top-m within its corresponding cluster and $h^{(u)} \in \mathcal{H}$ is the interactions of u in the original dataset.

4 Experiments

4.1 Experimental Settings

Datasets. To evaluate the performance of our method, we condense the training set of three real-world datasets: MIND [25], Goodreads [19], and MovieLens [11]. The split is 80%/10%/10% for train/validation/test. The condensation is conducted on the training

Table 1: Comparison between Condensed datasets and Original Datasets. We use "ORI.", "RD." and "MJ." to denote the original, random sampled, and majority-selected dataset.

Datasets	Methods	Item			User			Overall	
		avg.tok.	Size (KB)	Ratio	#users	Size (KB)	Ratio	Size (KB)	Ratio
MIND	OR.	45.42	10,384	100%	94,057	106,614	100%	116,998	100%
	RD.	16.73	4,095	39%	9,405	2,124	2%	6,219	5%
	MJ.	16.73	4,095	39%	9,405	5,733	5%	9,828	8%
	Ours	16.73	4,095	39%	9,405	2,139	2%	6,234	5%
Goodreads	OR.	35.37	2,189	100%	23,089	8,704	100%	10,893	100%
	RD.	16.01	993	45%	4,617	1,791	21%	2,784	26%
	MJ.	16.73	993	45%	2,350	1,977	23%	2,970	27%
	Ours	16.01	993	45%	4,617	1,957	22%	2,950	27%
MovieLens	OR.	34.80	232	100%	943	532	100%	764	100%
	RD.	15.26	121	52%	47	68	13%	189	25%
	MJ.	16.73	121	52%	47	232	44%	353	46%
	Ours	15.26	121	52%	47	81	15%	202	26%

Table 2: Overall Performance Comparison on Condensed Datasets. "Quality" denotes the achieved ratio of performance when compared to those trained on original datasets. The detailed condensation ratio could be found in Table 1.

Datasets		MIND[1], r=5%				Goodreads, r=27%				MovieLens, r=26%			
Rec Model	Metrics	Random	Majority	TF-DCon	*Original*	Random	Majority	TF-DCon	*Original*	Random	Majority	TF-DCon	*Original*
NAML	N@1	0.2871	0.2854	0.3071	*0.3176*	0.5197	0.5057	0.5411	*0.6462*	0.8241	0.8367	0.8484	*0.8280*
	N@5	0.3470	0.3466	0.3691	*0.3783*	0.7943	0.7884	0.8033	*0.8475*	0.8251	0.8397	0.8494	*0.8310*
	R@1	0.4016	0.4002	0.4377	*0.4534*	0.4520	0.4326	0.4704	*0.5635*	0.1785	0.1886	0.1873	*0.1752*
	R@5	0.5670	0.5697	0.6150	*0.6270*	0.9983	0.9986	0.9984	*0.9989*	0.7274	0.7323	0.7475	*0.7399*
	Quality	90.28%	90.15%	**97.22%**	*100.00%*	88.57%	87.01%	**90.49%**	*100.00%*	99.75%	102.18%	**103.15%**	*100.00%*
NRMS	N@1	0.2625	0.2631	0.2997	*0.3009*	0.5399	0.5094	0.5453	*0.6439*	0.8105	0.8294	0.8149	*0.8178*
	N@5	0.3225	0.3225	0.3597	*0.3608*	0.8017	0.7901	0.8054	*0.8476*	0.8227	0.8334	0.8371	*0.8253*
	R@1	0.3750	0.3793	0.4279	*0.4325*	0.4704	0.4344	0.4751	*0.5629*	0.1729	0.1821	0.1798	*0.1757*
	R@5	0.5414	0.5445	0.6000	*0.6042*	0.9982	0.9990	0.9985	*0.9993*	0.7325	0.7339	0.7446	*0.7321*
	Quality	88.23%	88.66%	**99.38%**	*100.00%*	90.47%	87.37%	**91.01%**	*100.00%*	99.31%	**101.57%**	101.28%	*100.00%*
Fastformer	N@1	0.2815	0.2736	0.3022	*0.3057*	0.5420	0.5165	0.5548	*0.6556*	0.7886	0.8105	0.8251	*0.7915*
	N@5	0.3425	0.3350	0.3637	*0.3645*	0.8028	0.7934	0.8092	*0.8529*	0.8254	0.8318	0.8429	*0.8145*
	R@1	0.3944	0.3804	0.4334	*0.4365*	0.4725	0.4414	0.4851	*0.5745*	0.1738	0.1799	0.1836	*0.1660*
	R@5	0.5631	0.5515	0.6096	*0.6144*	0.9983	0.9991	0.9986	*0.9990*	0.7390	0.7373	0.7465	*0.7279*
	Quality	92.01%	89.58%	**99.29%**	*100.00%*	89.74%	87.16%	**90.97%**	*100.00%*	101.80%	103.55%	**105.22%**	*100.00%*

sets of datasets and the test sets are from the original datasets.

Baselines. Due to the limited studies on condensation for CBR, we compare TF-DCon with two baselines implemented by ourselves: (i) *Random*. In Table 2, we randomly select a certain portion of users and randomly select a certain portion of tokens to be the contents for each item. (ii) *Majority*. We implement this method by sampling the users with a large number of interactions and compressing the contents of items by randomly sampling the tokens. For fair comparison, we maintained an equivalent number of synthesized users as employed in our method.

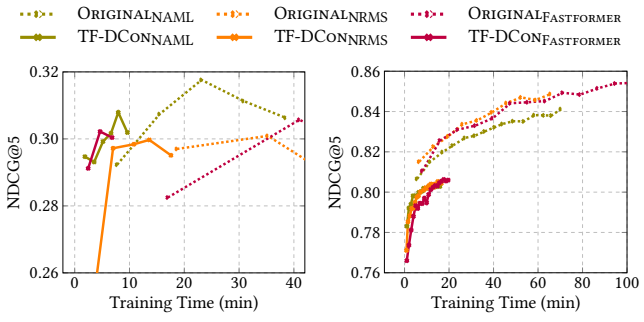
Evaluation Protocol. To evaluate our method, we test the performance of recommendation models trained on the condensed datasets. In this evaluation, we employ three prevalent content-based recommendation models, i.e., NAML [22], NRMS [23] and Fastformer [24]. Specifically, the evaluation is threefold: (1) condense the datasets, (2) train the recommendation models on the condensed datasets, and (3) test the performance of the models. We adopt the widely used metrics, i.e., NDCG@K and Recall@K, abbreviated as N@K and R@K. In this work, we set $K = 1, 5$ for Goodreads and MovieLens, and set $K = 5, 10$ for MIND. To compare the performance of models trained on condensed datasets and original datasets, we average the percentages of metrics relative to those with original datasets, denoting the percentages by "Quality".

Implementation Details. During condensation, we utilize GPT-3.5 for condensing. During training, we employ Adam [14] operator with a learning rate of 5e-3. For the backbone models, we set the content encoder dimension to 256, the user encoder dimension to 64, and the negative sampling ratio to 4. We average the results of five independent runs for each model. All experiments are conducted on a single NVIDIA GeForce RTX 3090 device.

4.2 Overall Comparison

To evaluate TF-DCon, we measure the performance of CBR models trained on the condensed data (Table 2) and their training efficiency (Figure 2). Further, the condensed data statistics are in Table 1.

Overall Performance Comparison. The proposed TF-DCon achieves better performance than the two baselines on three datasets. We utilize "'Quality" to evaluate the quality of condensed dataset,



(a) MIND **(b) Goodreads**

Figure 2: Train Efficiency on Original and Condensed Data.

which is computed by dividing each of the four metrics by the original performance, and then averaging them. Notably, as shown in Table 2, we approximate over 97% of the original performance with only 5% data on the MIND dataset across three different recommender models. From Table 1, we observe that the amazing condensation ratio comes from two parts: reducing the size of contents by 61% (item side) and reducing the size of historical interactions and users (user side) by 98%. We observe similar performance on datasets Goodreads and MovieLens. Interestingly, with MovieLens, we may achieve a slightly better performance than those trained on the original dataset. We argue that the condensed datasets may serve as a refined and denoised alternative to the original one. From Table 1, the size of the user part (i.e., users' interactions) is much larger than the size of the item part (item contents). Therefore, the predominant size reduction stems from the user part condensation.

Training Efficiency. Here, we compare the training time and performance of NAML, NRMS and Fastformer when trained on the original datasets and condensed datasets by TF-DCon. As shown in Figure 2, the solid lines represent the training curves on condensed datasets and the dotted line represents original datasets, where we can observe that the training on condensed datasets converges much faster than the training on original datasets (i.e., up to 5× speedup). Typically, the model convergences on condensed datasets are reached with around 6 minutes while the training on original datasets needs around or more than 30 minutes. For the performance, those trained on condensed datasets is comparable.

5 Conclusion

We introduce a novel method condensation method TF-DCon for content-based recommendation, where a prompt-evolving module is proposed to adapt ChatGPT to condense contents of items, and users and their corresponding historical interactions are condensed via a curated clustering-based synthesis module. This work represents the first exploration of dataset condensation for content-based recommendation and the first non-iterative synthetic data optimization approach. The experimental results on multiple real-world datasets verify the effectiveness of our proposed methods.

Acknowledgments

Jiahao Wu, Shengcai Liu, and Ke Tang would like to acknowledge the support provided by the National Key Research and Development Program of China under Grant No.2022YFA1004102, and in part by Guangdong Major Project of Basic and Applied Basic Research.

Wenqi Fan and Qing Li would like to acknowledge the support provided by Hong Kong Research Grants Council through Research Impact Fund (project no. R1015-23).

References

[1] Fares Antaki, Samir Touma, et al. 2023. Evaluating the performance of chatgpt in ophthalmology: An analysis of its successes and shortcomings.. In *medRxiv*.
[2] James RA Benoit. 2023. ChatGPT for Clinical Vignette Generation, Revision, and Evaluation.. In *medRxiv*.
[3] Tom Brown, Benjamin Mann, et al. 2020. Language models are few-shot learners. In *In Proc. of NeurIPS'2020*.
[4] Jonathan H Choi, Kristin E Hickman, Amy Monahan, and Daniel Schwarcz. 2023. Chatgpt goes to law school. In *Available at SSRN (2023)*.
[5] Aakanksha Chowdhery, Sharan Narang, et al. 2022. PaLM: Scaling Language Modeling with Pathways. arXiv:2204.02311
[6] Jiaxi Cui, Zongjian Li, et al. 2023. ChatLaw: Open-Source Legal Large Language Model with Integrated External Knowledge Bases. arXiv:2306.16092
[7] Haixing Dai, Zhengliang Liu, et al. 2023. AugGPT: Leveraging ChatGPT for Text Data Augmentation. *arXiv:2302.13007* (2023).
[8] Sunhao Dai, Ninglu Shao, et al. 2023. Uncovering ChatGPT's Capabilities in Recommender Systems. In *Proc. of RecSys'2023*.
[9] Wenqi Fan, Zihuai Zhao, et al. 2023. Recommender Systems in the Era of Large Language Models (LLMs). arXiv:2307.02046
[10] Xudong Han, Aili Shen, et al. 2022. Towards Fair Dataset Distillation for Text Classification. In *In Proc. of SustaiNLP'2022*. ACL.
[11] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* (2015).
[12] Wei Jin, Xianfeng Tang, et al. 2022. Condensing Graphs via One-Step Gradient Matching. In *Proc. of KDD'2022*. ACM.
[13] Wei Jin, Lingxiao Zhao, et al. 2022. Graph Condensation for Graph Neural Networks. In *Proc. of ICLR'2022*. OpenReview.net.
[14] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *In Proc. of ICLR'2015*.
[15] 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 Seventeen ACM International Conference on Web Search and Data Mining*.
[16] Qijiong Liu, Jieming Zhu, Lu Fan, Zhou Zhao, and Xiao-Ming Wu. 2024. STORE: Streamlining Semantic Tokenization and Generative Recommendation with A Single LLM. *arXiv preprint arXiv:2409.07276* (2024).
[17] Ilia Sucholutsky and Matthias Schonlau. 2021. Soft-label dataset distillation and text dataset distillation. In *IJCNN 2021*.
[18] Solomon Ubani et al. 2023. ZeroShotDataAug: Generating and Augmenting Training Data with ChatGPT. *arXiv:2304.14334* (2023).
[19] Mengting Wan and Julian McAuley. 2018. Item recommendation on monotonic behavior chains. In *Proc. of RecSys'2018*.
[20] Hongwei Wang, Fuzheng Zhang, et al. 2018. DKN: Deep Knowledge-Aware Network for News Recommendation. In *In Proc. of WWW'2018*. ACM.
[21] Tongzhou Wang, Jun-Yan Zhu, et al. 2018. Dataset Distillation. *arXiv* (2018).
[22] Chuhan Wu, Fangzhao Wu, et al. 2019. Neural News Recommendation with Attentive Multi-View Learning. In *Proc. of IJCAI'2019*. ijcai.org.
[23] Chuhan Wu, Fangzhao Wu, et al. 2019. Neural News Recommendation with Multi-Head Self-Attention. In *In Proc. of EMNLP-IJCNLP'2019*. ACL.
[24] Chuhan Wu, Fangzhao Wu, et al. 2021. Fastformer: Additive Attention Can Be All You Need. *arXiv:2108.09084* (2021).
[25] Fangzhao Wu, Ying Qiao, et al. 2020. MIND: A Large-scale Dataset for News Recommendation. In *Proc. of ACL'2020*.
[26] Jiahao Wu, Wenqi Fan, et al. 2024. Condensing Pre-augmented Recommendation Data via Lightweight Policy Gradient Estimation. *TKDE* (2024).
[27] Jiahao Wu, Wenqi Fan, Jingfan Chen, Shengcai Liu, Qing Li, and Ke Tang. 2022. Disentangled Contrastive Learning for Social Recommendation. In *Proc. of CIKM'2022*. ACM.
[28] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. 2021. Dataset Condensation with Gradient Matching. In *Proc. of ICLR'2021*. OpenReview.net.
[29] Xin Zheng, Miao Zhang, and other. 2023. Structure-free Graph Condensation: From Large-scale Graphs to Condensed Graph-free Data. *Proc. of NeurIPS'2023*.