Neighborhood-Based Collaborative Filtering for Conversational Recommendation

Zhouhang Xie*
zhx022@ucsd.edu
University of California, San Diego
La Jolla, CA, USA

Junda Wu*
juw069@ucsd.edu
University of California, San Diego
La Jolla, CA, USA

Hyunsik Jeon*
hyjeon@ucsd.edu
University of California, San Diego
La Jolla, CA, USA

Zhankui He
zhh004@ucsd.edu
University of California, San Diego
La Jolla, CA, USA

Harald Steck
hsteck@netflix.com
Netflix Inc.
Los Gatos, CA, USA

Rahul Jha
rahuljha@netflix.com
Netflix Inc.
Los Gatos, CA, USA

Dawen Liang
dliang@netflix.com
Netflix Inc.
Los Gatos, CA, USA

Nathan Kallus
nkallus@netflix.com
Netflix Inc.
Los Gatos, CA, USA
Cornell University
New York, NY, USA

Julian McAuley
jmcauley@ucsd.edu
University of California, San Diego
La Jolla, CA, USA

ABSTRACT

Conversational recommender systems (CRS) should understand users' expressed interests, which are frequently semantically rich and knowledge-intensive. Prior works attempt to address this challenge by using external knowledge bases or parametric knowledge in large language models (LLMs). In this paper, we study a complementary solution, exploiting item knowledge in the training data. We hypothesize that many inference-time user requests can be answered by reusing popular crowd-written answers associated with similar training queries. Following this intuition, we define a class of neighborhood-based CRS that makes recommendations by identifying items commonly associated with similar training dialogue contexts. Experiments on Inspired, Redial, and Reddit-Movie benchmarks show our method outperforms state-of-the-art LLMs with 2 billion parameters, and offers on-par performance to 7 billion parameter models while using over 170 times less GPU memory. We also show neighborhood and model-based predictions can be combined to achieve further performance improvements[1].

ACM Reference Format:
Zhouhang Xie, Junda Wu, Hyunsik Jeon, Zhankui He, Harald Steck, Rahul Jha, Dawen Liang, Nathan Kallus, and Julian McAuley. 2024. Neighborhood-Based Collaborative Filtering for Conversational Recommendation. In *18th ACM Conference on Recommender Systems (RecSys '24), October 14–18, 2024, Bari, Italy*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3640457.3688191

*All authors contributed equally to this research.
[1]https://github.com/zhouhanxie/neighborhood-based-CF-for-CRS

RecSys '24, October 14–18, 2024, Bari, Italy

ACM ISBN 979-8-4007-0505-2/24/10
https://doi.org/10.1145/3640457.3688191

1 INTRODUCTION

Conversational Recommender Systems (CRS) can interact with users, understand user preferences expressed in natural language, and perform item recommendation [11, 36]. A core requirement of a CRS is to handle semantically rich queries (e.g., "`movies that featured luxurious ship travel in the early 1900s`"), as well as questions that are related to factual information about movies (e.g., "`the most hilarious Steven Seagel movie`"). To handle free-from natural language inputs, recent works in CRS typically train models that map dialogue contexts into dense representations and predict the compatibility of items based on such dialogue representations [1, 11, 20, 35].

However, a challenge for this class of methods is handling detailed features about items. In particular, fine-grained knowledge about entities is often long-tailed [15]. For example, there is no guarantee that characteristics of a movie such as "`featuring early 1900s luxurious ship travel`" will appear frequently enough in the training dataset for a machine learning model to connect such characteristics to associated items. To address such a challenge, prior work augments CRS with external knowledge such as knowledge graphs [1, 35] and item reviews [22] or adopts large language models (LLMs) and exploit their knowledge from pre-training [11]. However, auxiliary knowledge about products is not always available, and knowledge in LLM pre-training does not necessarily transfer to out-of-distribution domains. In this way, building generic CRS algorithms without knowledge bases or pre-trained knowledge in LLMs remains an open problem.

In this work, we explore a complementary solution, exploiting crowd knowledge in the CRS training data. We start from a simple intuition that other users have already addressed many inference-time user requests in these existing dialogues. Following such an intuition, we propose a class of neighborhood-based CRS (NBCRS) that combines the idea of retrieval augmented generation (RAG) [8, 16, 19] and neighborhood-based recommender systems [5]. Inspired by the success of nearest neighbor language

models [16] that ensembles a retrieval component and a language model, our method builds a data store by embedding training dialogue contexts into sentence representations and tracking their associated items. Given a dialogue context, NBCRS makes predictions by retrieving a neighborhood of similar training data and recommends consensus items in the neighborhood commonly mentioned by the crowd.

By relying on retrieval instead of a model's parameters for memorizing information in training data, our method sidesteps the challenge of representing item properties in model parameters. Prior works show such a paradigm is an effective technique for knowledge-intensive NLP tasks [8, 19]. Unlike RAG, where the retrieved knowledge typically describes definitive facts (e.g., "the capital of France is Paris"), our method aggregates the crowd's wisdom by recommending items most commonly agreed upon given a neighborhood of queries. This formulation allows our method to handle ambiguous questions such as "`the most hilarious movie,`" where various candidate items are factually correct answers.

Experiments on Redial [20], Inspired [36], and Reddit-Movie (Reddit) [11] benchmarks show that despite its simplicity, our method outperforms various baseline methods, including state-of-the-art LLMs with 2 billion parameters, and offers on-par performance compared to larger models with 7 billion parameters while consuming over 170 times less GPU memory. We then discuss opportunities for combining neighborhood and model-based CRS, and show training the retriever model as part of a model-based CRS is an effective method for improving neighborhood-retrieval quality. Meanwhile, the model-based predictor component for training the retriever can be used as an item reranker to enhance the neighborhood-based model. Finally, we show that neighborhood-based prediction can be applied to LLMs to improve their recommendation performance, especially on small datasets such as Inspired and smaller models such as Gemma-2B [28]. Our results highlight the promise of modeling long-tailed crowd-generated signals in training data for CRS.

2 NEIGHBORHOOD-BASED CRS

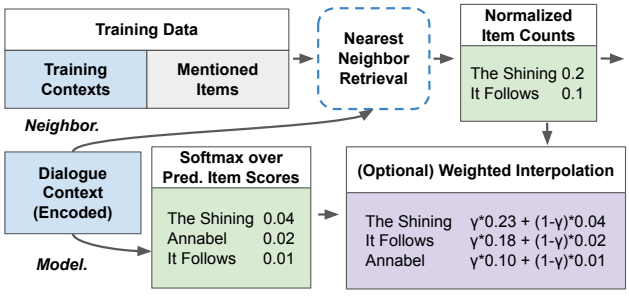


Figure 1: Overview of NBCRS. There are two components: a core `Neighrborhood-based` component (NB) that can be used as-is, and an optional `Model-based` component (MB) for fine-tuning purposes. The output from the two components can also be combined for improved performance.

We start by introducing the neighborhood-based CRS framework. As shown in Figure 1, NBCRS has two main components: a neighborhood-based component and an *optional* model-based recommender component. The neighborhood-based component retrieves a set of training contexts similar to the test-time context and recommends frequently mentioned items within each neighborhood. These item counts are then used to rank items as-is or normalized into a probability distribution. Conversely, the model-based component directly learns to score item compatibility based on the dialogue context. The results from the model-based and neighborhood-based components can be mixed together in different ways. For instance, in Figure 1, the raw scores from both components are normalized into two probability distributions and interpolated with a weighted factor γ into a final probability distribution. We chose this ensembling formulation in this work to study the relative importance of two components (Section 4). However, our framework is flexible and can accommodate other ensembling techniques.

Recommendation via Neighborhood Retrieval. Given a training dataset $\mathcal{D}$ that consists of $(\mathbf{q}, \tilde{\mathbf{v}})$ pairs of dialogue context $\mathbf{q}$ and associated items $\tilde{\mathbf{v}}$, f_{retrieve} requires a feature extraction function $\mathbf{\Phi}$ that maps each dialogue context into a vector representation. At inference time, the similarity of a training context and the test context can then be calculated via vector similarity function $\text{sim}(\cdot, \cdot)$. In this way, we can retrieve a neighborhood $\mathcal{N}$ of k most relevant training contexts and their associated items at inference time, and calculate the probability of items via normalized item counts in the neighborhood. Concretely, given a test context $\mathbf{q}$, the score of an item $\mathbf{v}$ given by the neighborhood-based component is calculated as:

$$\text{score}_{\text{neighbor}}(\mathbf{v}|\mathbf{q}) = \sum_{(\mathbf{q}_i, \tilde{\mathbf{v}}_i) \in \mathcal{N}} \mathbf{w_i}(\tilde{\mathbf{v}}_i = \mathbf{v}). \tag{1}$$

We could then rank candidate items by their score. Additionally, when obtaining the probability distribution over items is useful, these scores can be turned into item probabilities $\text{p}_{\text{neighbor}}$ using a softmax function or naive normalization. This formulation is similar to classical neighborhood-based recommender systems [5] and can flexibly incorporate various feature extraction functions, vector similarity functions, and normalization techniques.

The (Optional) Model-based Component. The neighborhood component can be used as-is using feature extraction functions that do not require training, such as pre-trained sentence embedding models and bag-of-word representations. We empirically show that using an off-the-shelf dense retriever as Φ already makes the neighborhood component perform competently (Section 4). However, it is unclear how to improve the retrieval quality, which are commonly addressed by fine-tuning the retriever in other information retrieval tasks. Inspired by recent work showing sentence representation can be improved simply via training language models that make prediction based on such representation (i.e., nearest neighbor language models) [16], we introduce a model-based component that encourages the dialogue context representation produced by a parametric retriever to be predictive of its associated items. In this way, dialogue contexts more likely to relate to similar items will be pulled closer in the representation space of Φ.

Concretely, when the feature extraction component Φ is a parametric encoder such as a sentence embedding model, we can introduce a parametric function g that maps a query and an item to its compatibility score:

$$\text{score}_{\text{model}}(\mathbf{v}|\mathbf{q}) = g(\phi(\mathbf{q}), \mathbf{v}). \tag{2}$$

The model can then be optimized so that compatible items will be assigned higher scores during training. For example, in this work, we maintain a learnable embedding for each item and implement the scoring function $\mathbf{g}(\mathbf{v}|\mathbf{q})$ as the dot product between a linear projection of the encoder output and the item embedding, following common practice in recommender systems [10]. Given an input dialogue context, the probability distribution over all items is then:

$$\text{p}_{\text{model}}(\mathbf{v}|\mathbf{q}) = \frac{\exp(\text{score}_{\text{model}}(\mathbf{v}|\mathbf{q}))}{\sum_j \exp(\text{score}_{\text{model}}(\mathbf{v_j}|\mathbf{q}))}. \tag{3}$$

We can thus optimize the neighborhood component using cross-entropy loss. This formulation allows different items to directly compete for probability, which has been shown to be effective in recommendation tasks [21]. However, we note that this is only one instance of the model-based component and our framework can flexibly include any method that learns to predict associated items given the query representation from the encoder Φ.

Combining Two Components for Inference. The main utility of the model-based component is to improve the quality of the feature extraction module Φ. However, the output from these two components can be combined to achieve further performance improvements. For example, let $\text{p}_{\text{neighbor}}$ be the softmax-normalized $\text{score}_{\text{neighbor}}$ over items given a query; we can obtain interpolated item probability from the two components as:

$$\mathbf{p}(\mathbf{v}|\mathbf{q}) = \gamma \mathbf{p}_{\text{neighbor}}(\mathbf{v}|\mathbf{q}) + (1-\gamma)\mathbf{p}_{\text{model}}(\mathbf{v}|\mathbf{q}). \tag{4}$$

This variant of NBCRS is related to nearest neighbor language models [16], which interpolates the probability of a neighborhood-based component and a language model. However, the core of NBCRS is the neighborhood-based component that exploits crowd-generated behavioral signal in the training data, via aggregating common answers for neighborhoods of dialogue contexts, rather than to generate text.

Dataset	Total Movies	N. Train Samples	N. Test Samples
Inspired	1506	731	211
Redial	6476	8929	4288
Reddit	29705	39928	19438

Table 1: Statistics of the Datasets

3 EXPERIMENTS

We experiment with NBCRS with various settings: (1) `Zero-Shot` that uses a pre-trained retriever for neighborhood-based prediction, (2) `NB` that uses the fine-tuned version of the same retriever (via `MB`) for neighborhood-based prediction, (3) `MB` that uses the model-based component for prediction (for reference), and (4) `N+M` that uses the model-based component only for reranking closely ranked items from `NB`. Note that this setting is equivalent to setting γ to be very close to 1 ($1 - 10^{-10}$ in practice) in Equation (4). We further discuss the effect of different weights of γ in Section 4.

Datasets and Evaluation. We conduct our experiments on Redial [20], Inspired [9], and Reddit-Movie (Reddit) [11] datasets. Dataset statistics are shown in Table 1. We use the same test splits as recent work [11], held out 20% of the training set for validation, and evaluate models' performance by Recall@k where $k \in \{1, 5, 20\}$.

Baselines. We compare NBCRS with open-source LLMs of various sizes, namely Gemma[28]-2B and Vicuna [2]-7B, as recent works show these LLMs are the state-of-the-art CRS on datasets studied in this work [11]. In our initial exploration, we find Vicuna-7B has better performance than Gemma-7B on Reddit and choose Vicuna as our 7B baseline. We also compare our model with a representative collaborative filtering model (FISM [14]) and popularity-based recommendation (PopRec). Finally, while the primary focus of this work is to develop CRS models that does not assume availability of knowledge-bases, we additionally include two recent knowledge-base-grounded CRS models (KGSF [35] and UniCRS [31]) to better understand to what extent models relying purely on collaborative signals in training data can match the performance of models enhanced by structured knowledge bases..

Implementation Details. We use PyTorch [26] for all of our implementations. For NBCRS, we use cosine similarity for retrieval, and set the weight of each observed item occurrence in the neighborhood $w_i = 1$. For sentence encoder ϕ we use a BERT-based [29] pre-trained sentence encoder[2] by default. We tune the number of neighbors for NBCRS between $\{1, 5, 30, 60, 90, 120, 150\}$ using the validation sets. For the item embedding for the `MB` component, we tune the dimension between $\{8, 16, 32\}$ latent factors on Reddit and use 16 factors across experiments. Finally, we follow [11] for implementing LLM-based methods, instructing the models to generate lists of items based on dialogue context, and tune the LLMs using low-rank adaption [13] using the same input and output format as in [11]. We run all GPU-requiring experiments on Nvidia RTX A6000s with 48GB memory.

4 RESULTS AND ANALYSIS

General Performance. Our main results are shown in Table 2. Notably, the `Zero-Shot` variant of NBCRS already outperforms zero-shot or fine-tuned Gemma-2B while achieving comparable performance to vicuna-7B. Further, retrieving training samples using a fine-tuned retriever (`NB`) outperforms `Zero-Shot`, showing that `MB` is effective at improving retrieval quality. Such improvement persists even when the dataset is too small for `MB` to converge: for example, the model-based component has low performance on Inspired, but it still improves the performance of `NB`. Meanwhile, using the model-based component for reranking items (`N+M`) with similar scores from `NB` consistently improves performance. Finally, we note that NBCRS is parameter efficient compared to LLMs, and show parameter-counts (*including the data store*) w.r.t. performance on Reddit for various models in Figure 2d. To this end, NBCRS has a much smaller memory footprint than LLMs, resulting in over 170

[2]https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Model	Setting	Inspired			Reddit			Redial		
		Recall@1	Recall@5	Recall@20	Recall@1	Recall@5	Recall@20	Recall@1	Recall@5	Recall@20
KGSF	Sft+KG	1.73 0.71	3.99 1.19	9.17 0.39	0.32 0.00	2.75 1.78	8.90 0.49	2.35 0.03	7.64 0.00	17.05 0.01
UniCRS	Sft+KG	2.04 0.32	8.03 0.94	18.59 0.12	0.97 0.13	3.44 0.23	9.79 0.50	4.09 0.08	12.80 0.07	27.12 0.28
Popularity	-	0.0 0.00	6.6 1.71	11.3 2.19	0.18 0.03	0.60 0.05	2.21 0.10	0.0 0.00	1.35 0.17	6.01 0.36
FISM	Sft	1.89 0.10	6.37 0.40	13.45 0.49	**1.93** 0.07	3.65 0.24	6.51 0.48	0.85 0.09	3.40 0.52	8.24 0.46
Gemma-2B	Zero-Shot	1.42 0.82	2.84 1.15	4.74 1.47	0.44 0.05	1.77 0.09	2.89 0.12	1.19 0.16	3.66 0.29	5.78 0.36
	Sft	0.0 0.00	0.95 0.67	2.37 1.05	0.54 0.05	2.13 0.10	3.49 0.13	1.63 0.19	4.13 0.30	5.11 0.33
Vicuna-7B	Zero-Shot	3.32 1.24	**8.53** 1.93	11.37 2.20	1.13 0.07	3.89 0.14	6.06 0.17	3.10 0.26	9.09 0.44	13.67 0.52
	Sft	**3.79** 1.31	8.06 1.88	10.43 2.11	1.21 0.08	4.39 0.15	7.18 0.18	**3.29** 0.27	**9.14** 0.44	13.67 0.52
NBCRS	Zero-Shot	0.47 0.47	4.73 1.46	14.69 2.44	1.28 0.08	5.56 0.16	14.08 0.24	1.46 0.18	6.64 0.38	16.34 0.56
	NB	1.42 1.04	6.63 1.77	**16.11** 2.23	1.24 0.07	5.94 0.16	15.52 0.25	1.23 0.16	6.20 0.37	16.58 0.54
	MB	0.0 0.00	0.0 0.00	1.42 0.81	1.14 0.07	5.17 0.15	13.62 0.24	1.25 0.17	5.13 0.33	14.80 0.54
	N+M	1.42 0.11	6.63 0.17	15.16 0.21	1.26 0.08	**5.95** 0.16	**15.58** 0.26	1.46 0.16	6.25 0.36	**16.86** 0.57

Table 2: Performance of models across datasets with standard errors. The reported numbers are percentages. Best performance excluding and including knowledge-graph-enhanced models are bolded and underlined, respectively.

Dataset	Gemma-2B			Vicuna-7B		
	R@1	R@5	R@20	R@1	R@5	R@20
Inspired	**3.79** 166%	**10.90** 283%	**14.22** 200%	3.32 -12%	**9.95** 16.6%	**13.74** 20.8%
Reddit	**1.40** 159%	**5.41** 153%	**13.33** 281%	**1.39** 14%	**4.74** 8%	**12.18** 69%
Redial	**2.56** 57%	**9.21** 123%	**16.98** 193%	2.91 -11%	8.29 -9%	**16.65** 21%

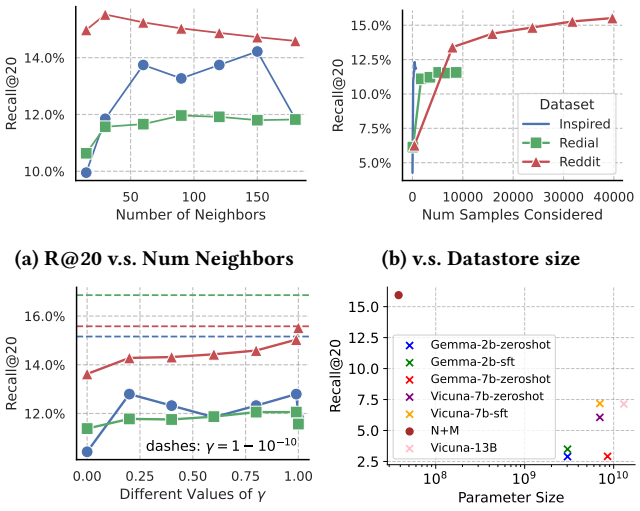
Table 3: Performance and percent improvement from using LLM's internal representation NB compared to generating recommended items from zero-shot or fine-tuned LLM, whichever is better.

times less GPU usage. Meanwhile, compared to Vicuna-7B with a batch size of 1, our method requires 0.06 seconds per sample, offering over 100 times inference speed-up.

Neighborhood-based Signals from MB *v.s. Content-based CRS with LLMs.* We conduct an ablation experiment where we take the last hidden state representation from zero-shot LLMs for retrieval using the NB framework and report its performance compared to directly generating recommendations using the LLMs. As shown in Table 3, NB with LLMs outperforms directly generating items. This is because NB is a form of collaborative filtering that aggregates crowd-answers, whereas LLMs focus on content-based recommendation [11, 24]. This observation highlights the importance of behavioral signals in CRS. The results also show NBCRS is not at odds with LLMs and can be applied to LLMs for improved performance.

Neighborhood-based v.s. Knowledge-base-enhanced Models. Compared to UniCRS and KGSF, NBCRS has stronger performance on the Reddit-Movie benchmark, while UniCRS and KGSF have stronger performance on Inspired and Redial. We hypothesize that such a difference in performance across datasets is because these benchmarks differ in their characteristics. In particular, Inspired and Redial are collected by asking crowd-workers to converse based on a fixed set of items and thus contain clean linkage of items to KGs, while the Reddit-Movie benchmark is constructed via collecting larger-scale, naturally occurring conversations online and thus contains semantically richer user requests [11]. To this end, our results show NBCRS is a complementary solution to existing methods for handling complex and noisy user queries at scale, while knowledge-enhanced CRS models have better performance when structured knowledge of items is available, and entities in conversation data can be cleanly linked to the associated knowledge base.

How Does Neighborhood Size Affect Performance? We report the effect of the number of neighbors on performance for NB in Figure 2a. As shown, the optimal values of k are dataset-dependent. However, the optimal k on the Reddit-Movie dataset is the smallest, likely because Reddit contains more complex queries [11], and thus, a smaller number of neighbors allows for more accurate answers.



(a) R@20 v.s. Num Neighbors **(b) v.s. Datastore size**

(c) v.s. Neighborhood Weight **(d) v.s. Model Size (on Reddit)**

Figure 2: Analysis and ablations

How Does Datastore Size Affect Performance? We report the performance of fine-tuned neighborhood-based model using 30 neighbors w.r.t. datastore size in Figure 2b. As shown in the figure, the neighborhood-based component benefits from larger data stores and the trend for improvement continues beyond using all available training data.

On the Importance of Neighborhood-based and Model Based Components. Since our proposed framework can combine both neighborhood-based and model-based components, a natural question is which component is more important. To this end, we report the performance of N+M models with $\gamma \in \{0.2 * t | t = 0, 1, ..., 5\} \cup 0.99$ (dots in the figure) in Figure 2c, where larger γ places more weights on NB. We additionally report the performance of the default N+M model that assigns a small weight $\gamma = 1 - 10^{-10}$ to MB. Interestingly, assigning a minuscule weight to γ achieves the best performance. The results also show that MB can indeed learn to predict connections that cannot be captured via retrieval, yet the accuracy of such prediction yields limited benefit compared to NB.

5 RELATED WORK

Conversational Recommendation. CRS can understand user interests and provide relevant recommendations [1, 3, 11, 12, 17, 18, 20, 31, 33]. To enhance the item knowledge of CRS, prior works typically incorporate external resources, such as symbolic knowledge bases [1, 35], reviews [23], or parametric knowledge in LLMs [11]. In contrast, our work exploits behavioral signals in the training dataset via retrieving dialogue histories and aggregating common associated items, which exploits crowd-wisdom and dooes not make any assumption on the availability of item knowledge graphs or item information in LLMs' parametric knowledge. Such a paradigm is complementary to content-based signals from auxiliary item knowledges in prior works.

Neighborhood Based Recommender System. Neighborhood-based methods are classical methods in recommender systems [4, 5, 25, 27, 30] that make predictions based on existing user interactions stored in the system. Recent studies [7] show that neighborhood-based methods still compete with neural network models, but their effectiveness in CRS has yet to be studied. To this end, our work demonstrates the effectiveness of neighborhood-based signals in conversational recommendation.

Retrieval Augmented Methods in NLP. Another related line of work to NBCRS is retrieval augmented methods that are known to improve model performance in various NLP tasks [8, 16, 19]. While recent proposals argue retrieval-augmentation is a promising direction in CRS [6], the effectiveness of these frameworks has not been studied, which this work addresses. Further, retrieval-augmented methods focus on content-based signals via retrieving *answers*, while our frameworks incorporate collaborative filtering (i.e., crowd-generated) signals by aggregating common answers from the crowd written for similar *questions*.

6 CONCLUSION

In this paper, we study the validity of a simple hypothesis: given a sufficiently large training dataset in conversational recommendation, many user requests can be addressed by reusing answers from similar previous requests. Following this insight, we show the empirical effectiveness of a neighborhood-enhanced conversational recommendation model, NBCRS. Despite being conceptually straightforward and compute-efficient, our method frequently outperforms previous state-of-the-art LLM-based CRS. Our results highlight the importance of modeling behavioral signals in the training data in addition to content-based signals from external item knowledge in conversational recommendation.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers and Yupeng Hou for their valuable insights. Implementations in this work benefited from CRSLab [34] and HuggingFace Transformers [32] libraries. This work is supported by a research award from Netflix.

REFERENCES

[1] Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. 2019. Towards Knowledge-Based Recommender Dialog System. 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)*. 1803–1813.

[2] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality. https://lmsys.org/blog/2023-03-30-vicuna/

[3] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 815–824.

[4] Joaquin Delgado and Naohiro Ishii. 1999. Memory-Based Weighted-Majority Prediction for Recommender Systems. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. https://api.semanticscholar.org/CorpusID:16989067

[5] Mukund Deshpande and George Karypis. 2004. Item-based top-N recommendation algorithms. *ACM Trans. Inf. Syst.* 22 (2004), 143–177. https://api.semanticscholar.org/CorpusID:207650042

[6] Dario Di Palma. 2023. Retrieval-augmented Recommender System: Enhancing Recommender Systems with Large Language Models. In *Proceedings of the 17th ACM Conference on Recommender Systems* (Singapore, Singapore) *(RecSys '23)*. Association for Computing Machinery, New York, NY, USA, 1369–1373. https://doi.org/10.1145/3604915.3608889

[7] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. 2019. Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems* (Copenhagen, Denmark) *(RecSys '19)*. Association for Computing Machinery, New York, NY, USA, 101–109. https://doi.org/10.1145/3298689.3347058

[8] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*. JMLR.org, Article 368, 10 pages.

[9] Shirley Anugrah Hayati, Dongyeop Kang, Qingxiaoyang Zhu, Weiyan Shi, and Zhou Yu. 2020. INSPIRED: Toward Sociable Recommendation Dialog Systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 8142–8152.

[10] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[11] Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian Mcauley. 2023. Large Language Models as Zero-Shot Conversational Recommenders. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (, Birmingham, United Kingdom,) *(CIKM '23)*. Association for Computing Machinery, New York, NY, USA, 720–730. https://doi.org/10.1145/3583780.3614949

[12] Zhankui He, Handong Zhao, Tong Yu, Sungchul Kim, Fan Du, and Julian McAuley. 2022. Bundle MCR: Towards Conversational Bundle Recommendation. In *Proceedings of the 16th ACM Conference on Recommender Systems*. 288–298.

[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] Santosh Kabbur, Xia Ning, and George Karypis. 2013. FISM: factored item similarity models for top-N recommender systems. In *Proceedings of the 19th*

ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (Chicago, Illinois, USA) *(KDD '13)*. Association for Computing Machinery, New York, NY, USA, 659–667. https://doi.org/10.1145/2487575.2487589

[15] Nikhil Kandpal, H. Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2022. Large Language Models Struggle to Learn Long-Tail Knowledge. In *International Conference on Machine Learning*. https://api.semanticscholar.org/CorpusID:253522998

[16] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through Memorization: Nearest Neighbor Language Models. In *International Conference on Learning Representations*. https://openreview.net/forum?id=HklBjCEKvH

[17] Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. 2020. Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 304–312.

[18] Wenqiang Lei, Gangyi Zhang, Xiangnan He, Yisong Miao, Xiang Wang, Liang Chen, and Tat-Seng Chua. 2020. Interactive path reasoning on graph for conversational recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 2073–2083.

[19] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (, Vancouver, BC, Canada,) *(NIPS '20)*. Curran Associates Inc., Red Hook, NY, USA, Article 793, 16 pages.

[20] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. *Advances in neural information processing systems* 31 (2018).

[21] 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*. 689–698.

[22] Yu Lu, Junwei Bao, Yan Song, Zichen Ma, Shuguang Cui, Youzheng Wu, and Xiaodong He. 2021. RevCore: Review-Augmented Conversational Recommendation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, Online, 1161–1173. https://doi.org/10.18653/v1/2021.findings-acl.99

[23] Yu Lu, Junwei Bao, Yan Song, Zichen Ma, Shuguang Cui, Youzheng Wu, and Xiaodong He. 2021. RevCore: Review-Augmented Conversational Recommendation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 1161–1173.

[24] Sheshera Mysore, Andrew Mccallum, and Hamed Zamani. 2023. Large Language Model Augmented Narrative Driven Recommendations. In *Proceedings of the 17th ACM Conference on Recommender Systems* (Singapore, Singapore) *(RecSys '23)*. Association for Computing Machinery, New York, NY, USA, 777–783. https://doi.org/10.1145/3604915.3608829

[25] Atsuyoshi Nakamura and Naoki Abe. 1998. Collaborative Filtering Using Weighted Majority Prediction Algorithms. In *International Conference on Machine Learning*. https://api.semanticscholar.org/CorpusID:31573203

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. *PyTorch: an imperative style, high-performance deep learning library*. Curran Associates Inc., Red Hook, NY, USA.

[27] Badrul Munir Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *The Web Conference*. https://api.semanticscholar.org/CorpusID:8047550

[28] Gemma Team. 2024. Gemma: Open Models Based on Gemini Research and Technology. arXiv:2403.08295 [cs.CL]

[29] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Neural Information Processing Systems*. https://api.semanticscholar.org/CorpusID:13756489

[30] Koen Verstrepen and Bart Goethals. 2014. Unifying nearest neighbors collaborative filtering. In *Proceedings of the 8th ACM Conference on Recommender Systems* (Foster City, Silicon Valley, California, USA) *(RecSys '14)*. Association for Computing Machinery, New York, NY, USA, 177–184. https://doi.org/10.1145/2645710.2645731

[31] Xiaolei Wang, Kun Zhou, Ji-Rong Wen, and Wayne Xin Zhao. 2022. Towards Unified Conversational Recommender Systems via Knowledge-Enhanced Prompt Learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1929–1937.

[32] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Qun Liu and David Schlangen (Eds.). Association for Computational Linguistics, Online, 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6

[33] Yiming Zhang, Lingfei Wu, Qi Shen, Yitong Pang, Zhihua Wei, Fangli Xu, Bo Long, and Jian Pei. 2022. Multiple Choice Questions based Multi-Interest Policy Learning for Conversational Recommendation. In *Proceedings of the ACM Web Conference 2022*. 2153–2162.

[34] Kun Zhou, Xiaolei Wang, Yuanhang Zhou, Chenzhan Shang, Yuan Cheng, Wayne Xin Zhao, Yaliang Li, and Ji-Rong Wen. 2021. CRSLab: An Open-Source Toolkit for Building Conversational Recommender System. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, Heng Ji, Jong C. Park, and Rui Xia (Eds.). Association for Computational Linguistics, Online, 185–193. https://doi.org/10.18653/v1/2021.acl-demo.22

[35] Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. 2020. Improving conversational recommender systems via knowledge graph based semantic fusion. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 1006–1014.

[36] Kun Zhou, Yuanhang Zhou, Wayne Xin Zhao, Xiaoke Wang, and Ji-Rong Wen. 2020. Towards Topic-Guided Conversational Recommender System. In *Proceedings of the 28th International Conference on Computational Linguistics*. 4128–4139.