Parameter-Efficient Conversational Recommender System as a Language Processing Task

Mathieu Ravaut[1], Hao Zhang[1], Lu Xu[2], Aixin Sun[1], Yong Liu[1]

[1]Nanyang Technological University, Singapore
[2]Singapore University of Technology and Design, Singapore

Conversational Recommender Systems (CRS)

Conversational Recommender Systems (CRS) jointly generate a natural language response to the user (**conversation task**) and recommend a list of items (**recommendation task**).

CRS approaches can be roughly divided into two categories :

- *Attribute-based* CRS : collect user preference on items attributes.
- *Generation-based* CRS : acquire feedback from users through language and generate natural responses.

We are focusing on **generation-based** CRS in this work.

Challenges in CRS

CRS are challenging to build because item recommendation and language generation are two tasks of very different nature.

A long line of work relies on knowledge graphs to learn items representation [1, 24, 23]. Unfortunately, there are a few issues :

- Because they are learned separately, word representations and items representations are semantically misaligned.
- KG consist in an external source of knowledge, which may not be readily available in certain inference setups.
- This approach neglects rich text information available for items.

Unifying CRS through language models

The recent MESE [21] approach uses pre-trained language models to learn items representations, and integrates them within the language response, bypassing the need for knowledge graph. However, it still relies on several models (two DistilBERT [15] and a GPT-2 [14]).

Overall, there does not exist yet a ***truly unified*** CRS model :

- UniCRS [20] uses a language model and a knowledge graph, and requires three training stages.
- BARCOR [19] and RecInDial [18] train in a single stage, but still need both a language model and a knowledge graph.
- MESE [21] discards the knowledge graph and still trains in a single stage, but relies on several pre-trained language models.

Proposal

In this work, we push simplicity and unification to its finest and solve the CRS task with a **single pre-trained language model (LM) fine-tuned in a single stage, without using a knowledge graph (KG)**.

Besides, through parameter-efficient fine-tuning, we only update a small fraction of parameters.

Dialogue Modeling

Let $\boldsymbol{\mathcal{I}} = \{I_1, I_2, \ldots, I_{N_{\text{item}}}\}$ be the item database with N_{item} items.

Let $\boldsymbol{\mathcal{D}} = \{D_1, D_2, \ldots, D_{N_{\text{dial}}}\}$ be the dataset with N_{dial} dialogues.

Let $D = \{u_t\}_{t=1}^{n_{\text{utt}}}$ be a dialogue with n_{utt} utterances.
Conditioning on the dialogue history $D_t = \{u_i\}_{i=1}^{t-1}$, CRS predicts :

- The current utterance $u_t = \{w_j\}_{j=1}^{n}$, with n tokens.
- The set of recommended items $\boldsymbol{\mathcal{I}}_t$, which may be empty.

Utterances are produced by the *seeker* or the *recommender*.
CRS only predicts the *recommender* utterances.

W use a decoder-only Transformer LM enhanced with special tokens : "[ITEM]", "[SEP]", "[REC]" and "[REC_END]".

Items Representation

We use the LM for both dialogue response and items representation.

Each movie item is described with a text in the template "*Movie title [SEP] Actors [SEP] Director(s) [SEP] Genre(s) [SEP] Plot*".

We add an item head h_{item} and learnable pooling weight w to the LM. The j-th item representation is :

$$\boldsymbol{v}_j = h_{\text{item}}(w^T \cdot \boldsymbol{I}_j). \tag{1}$$

where $\boldsymbol{I}_j$ is the LM contextual representation of the description.

Context Representation

For each utterance of the context $D_t = \{u_i\}_{i=1}^{t-1}$ we obtain contextual representation with the LM : $\boldsymbol{u}_i = [\boldsymbol{c}_{i,1}, \ldots, \boldsymbol{c}_{i,n}]$.

Item names are replaced by the "[ITEM]" special token.

If the utterance is from the speaker, it becomes $\bar{\boldsymbol{u}}_i = \tilde{\boldsymbol{u}}_i = [\boldsymbol{v}_{\text{sep}}, \boldsymbol{v}_j, \boldsymbol{v}_{\text{sep}}, \boldsymbol{u}_i]$.

If it is from the recommender, it becomes $\bar{\boldsymbol{u}}_i = \tilde{\boldsymbol{u}}_i = [\boldsymbol{v}_{\text{rec}}, \boldsymbol{v}_j, \boldsymbol{v}_{\text{rec\_end}}, \boldsymbol{u}_i]$.

If there is no recommended item, it remains unchanged $\bar{\boldsymbol{u}}_i = \boldsymbol{u}_i$.

Dialogue representation is $\boldsymbol{D}_t = [\bar{\boldsymbol{u}}_1, \ldots, \bar{\boldsymbol{u}}_{t-1}, \boldsymbol{v}_{\text{rec}}]$, and we use the output of the last "[REC]" token, noted $\boldsymbol{d}_t$.

Recommendation task (1/2)

We use a **contrastive learning** approach [4, 13, 12] to bring closer the query $\boldsymbol{d}_t$ and the positive item $\boldsymbol{v}_p$; while pushing apart $\boldsymbol{d}_t$ and M sampled negative items $\{\boldsymbol{v}'_j\}_{j=1}^{M}$.

$$\mathcal{L}_{\text{recall}} = -\frac{1}{|\boldsymbol{\mathcal{D}}|} \sum_{D_t \in \boldsymbol{\mathcal{D}}} \log(\mathcal{E}_{D_t}). \quad (2)$$

where :

$$\mathcal{E}_{D_t} = \frac{e^{f(\boldsymbol{d}_t)^\top \odot \boldsymbol{v}_p}}{e^{f(\boldsymbol{d}_t)^\top \odot \boldsymbol{v}_p} + \sum_{(\boldsymbol{d}_t, \boldsymbol{v}'_j) \sim \boldsymbol{\mathcal{N}}} e^{f(\boldsymbol{d}_t)^\top \odot \boldsymbol{v}'_j}}, \quad (3)$$

where where f is a projection head MLP.

We **stop the gradients** of LM and only optimize the pooling (w) and MLP layers (h_{item}, f).

Recommendation task (2/2)

To refine item selection, we use a **re-ranking** approach.

We concatenate the context and all items, $[\boldsymbol{D}_t, \boldsymbol{v}_p, \boldsymbol{v}'_1, \ldots, \boldsymbol{v}'_M]$.

This input is fed into LM then MLP f, with attention mask blocking attention between items, yielding representations $[\boldsymbol{q}_p, \boldsymbol{q}_1, \ldots, \boldsymbol{q}_M]$.

Another MLP layer g is applied to compute the final item scores as $\boldsymbol{r} = [g(\boldsymbol{q}_p), g(\boldsymbol{q}_1), \ldots, g(\boldsymbol{q}_M)] = [r_p, r_1, \ldots, r_M]$.

Items are re-ranked through a cross-entropy loss :

$$\mathcal{L}_{\text{rerank}} = \frac{1}{|\boldsymbol{\mathcal{D}}|} \sum_{D_t \in \boldsymbol{\mathcal{D}}} f_{\text{XE}}(\boldsymbol{r}, \boldsymbol{Y}), \tag{4}$$

where $\boldsymbol{Y} = [1, 0, \ldots, 0]$ and f_{XE} denotes cross-entropy loss.

Response generation task

If u_t contains an item to be recommended, it is appended to the context :

$$\tilde{\boldsymbol{D}}_t = [\bar{\boldsymbol{u}}_1, \ldots, \bar{\boldsymbol{u}}_{t-1}, \boldsymbol{v}_{\text{rec}}, \boldsymbol{v}_p, \boldsymbol{v}_{\text{rec\_end}}]. \tag{5}$$

otherwise, $\tilde{\boldsymbol{D}}_t = [\bar{\boldsymbol{u}}_1, \ldots, \bar{\boldsymbol{u}}_{t-1}]$.

Response generation is optimized by the standard next-token prediction objective :

$$\mathcal{L}_{\text{gen}} = -\frac{1}{|\boldsymbol{\mathcal{D}}|} \sum_{D_t \in \boldsymbol{\mathcal{D}}} \frac{1}{n} \sum_{j=1}^{n} \log(p_\theta(w_j | w_{1:(j-1)}, \tilde{\boldsymbol{D}}_t). \tag{6}$$

Parameter-efficiency

We keep the backbone LM frozen, and instead add LoRA [7] layers to be updated.
This prevents catastrophic forgetting of the LM's text generation capability, while adapting it to the CRS task [8, 22].

The only learnable weights are : task-specific MLP layers f, g, h_{item}, pooling weights w, and the special tokens embeddings.

Our model is dubbed Parameter-Efficient Conversational Recommender System (**PECRS**).

Training

We train in a *singe-stage* end-to-end manner by minimizing the following loss :

$$\mathcal{L} = \alpha \times \mathcal{L}_{\text{recall}} + \beta \times \mathcal{L}_{\text{rerank}} + \gamma \times \mathcal{L}_{\text{gen}}, \tag{7}$$

During training :

- Sample M_{train} negative items, and **share them across losses** $\mathcal{L}_{\text{recall}}$ and $\mathcal{L}_{\text{rerank}}$ and **across batch items**.
- Append the ground-truth item to the dialogue context.

Inference

During inference :

- Encode every single item.
- Retrieve the closest $M_{\text{inference}}$ items to the dialogue query via $f(\boldsymbol{d}_t)^\top \odot \boldsymbol{v}_j$.
- Re-rank them and output the highest score one as prediction.
- Append the predicted item to the context.
- The presence of "[ITEM]" in the generated response assesses recommendation.

Experimental Setup

We apply PECRS to **movie** recommendation on ReDial [9] and INSPIRED [5] datasets.

For the backbone LM, we use GPT-2 (**PECRS-small**) and GPT-2-medium (**PECRS-medium**).

We train with AdamW and LR as $3e-5$, warming up one epoch.

We set $M_{\text{train}} = 150$ for training and $M_{\text{infer}} = 700$ for inference.

We balance losses with $\alpha = 0.15$, $\beta = 0.85$, and $\gamma = 1.0$.

Evaluation Setup

We measure **recommendation** performance with *Recall@K (R@K)* metric, taking $K \in \{1, 10, 50\}$ and *Unique*, the number of unique recommended items throughout the test set.

We measure **conversation** with *Perplexity (PPL)* (fluency), *Distinct@K (Dist@K)* with $K \in \{2, 3, 4\}$ (diversity), *F-1* score of the presence of "[ITEM]" (recommendation decision) and *ROUGE-K (RG-K)*, taking $K \in \{1, 2\}$ (closeness to the ground truth).

Recommendation Results

Model	Metadata			Model Properties			ReDial				INSPIRED			
	KG	**Reviews**	**Description**	**Extra Model**	**PEFT**	**Rounds**	**R@1**	**R@10**	**R@50**	**Unique**	**R@1**	**R@10**	**R@50**	**Unique**
ReDial (Li et al., 2018)	✗	✗	✗	✓	✗	3	2.4	14.0	32.0	_	_	_	_	_
KBRD (Chen et al., 2019)	✓	✗	✗	✓	✗	2	3.0	16.3	33.8	_	_	_	_	_
KGSF (Zhou et al., 2020a)	✓	✗	✗	✓	✗	3	3.9	18.3	37.8	_	_	_	_	_
KECRS (Zhang et al., 2022)	✓	✗	✗	✓	✗	2	2.3	15.7	36.6	_	_	_	_	_
BARCOR (Wang et al., 2022b)	✓	✗	✗	✓	✗	1	2.5	16.2	35.0	_	_	_	_	_
UniCRS (Wang et al., 2022c)	✓	✗	✗	✓	✓	3	5.1	22.4	42.8	_	**9.4**	**25.0**	**41.0**	_
RecInDial (Wang et al., 2022a)	✓	✗	✗	✓	✗	1	3.1	14.0	27.0	_	_	_	_	_
VRICR (Zhang et al., 2023b)	✓	✗	✗	✓	✗	3	5.7	25.1	41.6	_	_	_	_	_
RevCore (Lu et al., 2021)	✓	✓	✗	✓	✗	2	**6.1**	23.6	45.4	_	_	_	_	_
C^2-CRS (Zhou et al., 2022)	✓	✓	✗	✓	✗	2	5.3	23.3	40.7	_	_	_	_	_
MESE (Yang et al., 2022)	✗	✗	✓	✓	✗	1	5.6	**25.6**	**45.5**	_	4.8	13.5	30.1	_
PECRS-small	✗	✗	✓	✗	✓	1	4.7	20.8	40.5	463	5.4	16.1	33.3	34
PECRS-medium	✗	✗	✓	✗	✓	1	5.8	22.5	41.6	**634**	5.7	17.9	33.7	**72**

PECRS-medium is on par with previous leading approaches (RevCore [11], MESE [21]) for Recall@1 on ReDial.

Scaling up LM size improves recall and items diversity (Unique).

Conversation Results

Model	Reference-based			Reference-free			
	RG-1	RG-2	F-1	PPL	Dist@2	Dist@3	Dist@4
C^2-CRS	_	_	_	_	0.163	0.291	0.417
UniCRS	_	_	_	_	0.492	0.648	0.832
RecInDial	_	_	_	_	0.518	0.624	0.598
MESE	_	_	_	12.9	**0.822**	1.152	1.313
PECRS-small	36.28	14.77	86.04	9.89	0.745	1.462	2.132
PECRS-medium	**36.86**	**15.27**	**86.36**	**8.98**	0.820	**1.552**	**2.154**

PECRS-medium reaches SOTA generation capability on ReDial.

Aspect	MESE	PECRS-small	Tie
Fluency	28.00 (1.63)	**46.67** (5.91)	25.33 (6.24)
Relevancy	26.33 (2.62)	**46.00** (0.82)	27.67 (2.87)

Which is confirmed by a human evaluation for fluency and relevancy.

Comparison with LLMs

We compare against popular instruction-tuned LLMs used in zero-shot [16, 2] on INSPIRED :

Model	**Rec.**				**Conv.**	
	R@1	**R@10**	**R@50**	**Unique**	**RG-1**	**RG-2**
PECRS-small	5.4	**16.1**	**33.3**	**34**	**29.72**	**8.26**
Llama-2-7B-chat	**9.3**	*9.3*	*9.3*	26	19.88	2.88
Vicuna-1.5-7B	8.2	*8.2*	*8.2*	23	21.18	3.50

LLMs used in this fashion tend to always recommend among the same small subset of items.

It is not straightforward how to score multiple items with LLMs in zero-shot.

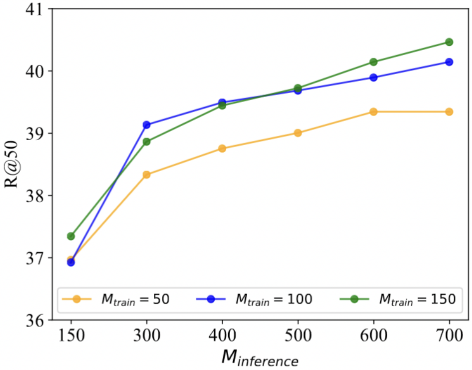
Analysis

Decoding Strategy	Reference-based		Reference-free		
	RG-1	RG-2	Dist@2	Dist@3	Dist@4
Greedy decoding	38.54	16.25	0.208	0.311	0.390
Beam search	38.23	16.83	0.235	0.353	0.444
Diverse beam search (diversity=0.5)	39.94	<u>17.30</u>	0.190	0.287	0.361
Diverse beam search (diversity=1.0)	**40.29**	**17.40**	0.179	0.264	0.320
Diverse beam search (diversity=1.5)	<u>40.07</u>	17.23	0.172	0.246	0.290
Top-k sampling (k=25)	33.54	14.40	0.593	1.177	1.806
Top-k sampling (k=50)	33.37	14.17	**0.647**	<u>1.300</u>	<u>1.989</u>
Top-k sampling (k=75)	33.48	14.15	<u>0.644</u>	**1.303**	**1.992**
Nucleus sampling (p=0.90)	36.35	16.04	0.329	0.555	0.760
Nucleus sampling (p=0.95)	36.44	16.02	0.351	0.594	0.804
Nucleus sampling (p=0.99)	36.60	16.07	0.352	0.593	0.809

Different decoding methods [17, 3, 6] yield very inconsistent Dist@K results.

We advocate for using reference-based methods like ROUGE [10], which are much more stable.

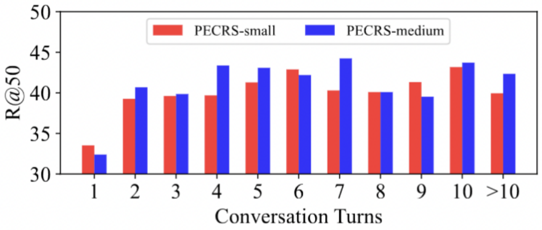
Analysis



The M parameter controlling the number of negatives is crucial.

A higher M is better, albeit at greater computational cost.

Analysis



Beyond the first turn, recall is relatively stable w.r.t the number of turns in the context.

Conclusion

In brief, we have introduced PECRS, a simple model fine-tuning a pre-trained LM in a single stage for the CRS task.

- Our model uses GPT-2 for both response generation and item encoding. This is rendered possible through :
  - Projection heads for items and items re-ranking.
  - Stop gradient operator on the backbone.
  - Parameter-efficiency LoRA.
- Optimization is streamlined through re-using the same negative samples across batch items and losses.
- For conversation evaluation, we advocate for not using the popular Dist@K metrics, and use reference-based metrics instead.

[1] Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. 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)*, pages 1803–1813, Hong Kong, China, November 2019. Association for Computational Linguistics.

[2] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna : An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https ://vicuna. lmsys. org (accessed 14 April 2023)*, 2023.

[3] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 889–898. Association for Computational Linguistics, July 2018.

[4] Michael U. Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications

to natural image statistics. *J. Mach. Learn. Res.*, 13 :307–361, feb 2012.

[5] Shirley Anugrah Hayati, Dongyeop Kang, Qingxiaoyang Zhu, Weiyan Shi, and Zhou Yu. INSPIRED : Toward sociable recommendation dialog systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8142–8152, Online, November 2020. Association for Computational Linguistics.

[6] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020.

[7] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA : Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[8] Zhiqiang Hu, Yihuai Lan, Lei Wang, Wanyu Xu, Ee-Peng Lim, Roy Ka-Wei Lee, Lidong Bing, Xing Xu, and Soujanya Poria. Llm-adapters : An adapter family for parameter-efficient fine-tuning of large language models. *ArXiv*, abs/2304.01933, 2023.

[9] Raymond Li, Samira Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. Towards deep conversational recommendations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, page 9748–9758. Curran Associates Inc., 2018.

[10] Chin-Yew Lin. ROUGE : A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[11] Yu Lu, Junwei Bao, Yan Song, Zichen Ma, Shuguang Cui, Youzheng Wu, and Xiaodong He. RevCore : Review-augmented conversational recommendation. In *Findings of the Association for Computational Linguistics : ACL-IJCNLP 2021*, pages 1161–1173, Online, August 2021. Association for Computational Linguistics.

[12] Andriy Mnih and Koray Kavukcuoglu. Learning word embeddings efficiently with noise-contrastive estimation. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[13] Andriy Mnih and Yee Whye Teh. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, page 419–426, 2012.

[14] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8) :9, 2019.

[15] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert : smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

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

[17] Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search : Decoding diverse solutions from neural sequence models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[18] Lingzhi Wang, Huang Hu, Lei Sha, Can Xu, Daxin Jiang, and Kam-Fai Wong. RecInDial : A unified framework for conversational recommendation with pretrained language models. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, pages 489–500, Online only, November 2022. Association for Computational Linguistics.

[19] Ting-Chun Wang, Shang-Yu Su, and Yun-Nung Chen. Barcor : Towards a unified framework for conversational recommendation systems. *ArXiv*, abs/2203.14257, 2022.

[20] Xiaolei Wang, Kun Zhou, Ji-Rong Wen, and Wayne Xin Zhao. Towards unified conversational recommender systems via knowledge-enhanced prompt learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, page 1929–1937. Association for Computing Machinery, 2022.

[21] Bowen Yang, Cong Han, Yu Li, Lei Zuo, and Zhou Yu. Improving conversational recommendation systems' quality with context-aware item meta-information. In *Findings of the*

Association for Computational Linguistics : NAACL 2022, pages 38–48, Seattle, United States, July 2022. Association for Computational Linguistics.

[22] Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. Llama-adapter : Efficient fine-tuning of language models with zero-init attention. *ArXiv*, abs/2303.16199, 2023.

[23] Tong Zhang, Yong Liu, Boyang Li, Peixiang Zhong, Chen Zhang, Hao Wang, and Chunyan Miao. Toward knowledge-enriched conversational recommendation systems. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 212–217, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[24] Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. Improving conversational recommender systems via knowledge graph based semantic fusion. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 1006–1014. Association for Computing Machinery, 2020.