Synergizing In-context and Supervised Learning for End-to-end Task-oriented Dialog Systems

Anonymous ACL submission

Abstract

Black-box language models (BLMs), large language models accessible only via an API, showcase remarkable (few shot) in-context learning performance for many NLP tasks. Our work explores their performance for end-to-end task-oriented dialog (TOD) systems, in the setting where a reasonable-sized training data is available. Benchmarking two BLMs (OpenAI's *ChatGPT* and *gpt-4*) on two end-to-end TOD datasets (MultiWoZ and SMD), we find that their performance is not on par with existing supervised SoTA models. In response, we propose *SincTOD*, which synergizes trained models with BLMs for superior performance. At a high level, SincTOD uses supervised models to provide additional hints and exemplar selection for BLM's in-context prompts. We show that *SincTOD* with *gpt-4* outperforms SoTA baselines on both datasets. Further, *SincTOD* also showcases strong performance in low-data setting, where it can be trained with less than 300 dialogs.

1 Introduction

Recent times have seen unprecedented progress in the field of NLP, through the rapid development and widespread use of extremely large language models (Bubeck et al., 2023; Hoffmann et al., 2022; Google, 2023; Touvron et al., 2023; ope, 2022). Of these, some of the largest (and best performing) models do not release their parameters publicly and are only accessible through an API call. We call such models *black-box language models* (BLMs).

BLMs such as *ChatGPT* and *gpt-4* have shown remarkable performance in various NLP tasks, especially in zero and few shot settings. These include question answering (Google, 2023), reasoning (bench authors, 2023), summarization (Pu et al., 2023; Zhang et al., 2023), and our focus, task oriented dialog (TOD) (Hudecek and Dusek, 2023; Hu et al., 2022). However, to the best of our knowledge, no work has studied them in the context of *end-to-end TOD*, i.e., setting where no intermediate supervision is available for TOD training.

Most existing works apply BLMs in a zero-shot or few-shot setting via in-context learning but do not explore their applicability when a reasonable amount of training data is available for the task. In our preliminary work, we find that BLMs coupled with standard few-shot in-context learning do not match up to the state-of-the-art supervised performance for popular end-to-end TOD datasets, such as MultiWoZ (Budzianowski et al., 2018) and SMD (Eric et al., 2017).

Our paper asks the following question: *can BLMs contribute to pushing the state of the art in end-to-end supervised TOD?* In response, we propose *SincTOD*, which synergizes supervised models with BLMs for superior performance. *SincTOD* leverages training data to build auxiliary models that predict hints, such as the types of entities expected in the response, dialog closure, and response size. Predicted hints are used first to select quality exemplars and are systematically incorporated into the BLM prompts. We find that our hint-augmented prompts lead BLMs to generate superior responses than SoTA supervised models for both datasets.

We additionally experiment in settings where amount of training data is limited. There, *SincTOD*'s gains are even more salient. Overall, our experiments suggest that while BLMs may have a role to play in supervised settings, it may necessitate a careful task-specific design to combine trained models and BLMs for better performance.

2 Related Works

Conventional TOD systems follow the modular design (Young et al., 2013; Rojas-Barahona et al., 2016; Hosseini-Asl et al., 2020) and require annotations for natural language understanding, dialog state tracking, and response generation modules. This work, however, focuses on end-to-end TOD systems (Eric et al., 2017; Madotto et al., 2018; Wu et al., 2019) that alleviate the need for annotations by directly predicting the response given dialog history and KB.

Though BLMs have been explored for TOD tasks (Hu et al., 2022; Hudecek and Dusek, 2023; Bang et al., 2023; Li et al., 2023), to the best of our knowledge, we are the first to explore them in an end-to-end setting. Directional Stimulus Prompting (DSP), an approach closer to ours, uses keywords and dialog acts as hints for summarization and response generation tasks, respectively (Li et al., 2023). However, unlike DSP, *SincTOD* uses multiple hints – entity types, dialog closure, and response size – relevant to the TOD task. Further, *SincTOD* uses these hints to also improve the quality of the in-context exemplars. Finally, *SincTOD* prompt is carefully designed to nudge BLM towards the desired reasoning behavior.

3 *SincTOD*

Let $c = [u_1, s_1, u_2, s_2, ..., u_i]$ be a dialog context with i turns where u and s denote user and system utterances respectively. In addition, we have a knowledge base (KB) K associated with the user goal. A TOD system's task is to predict the follow-up response s given (c, K). In the end-to-end setting, a TOD system is learned solely over a dataset $\mathcal{D} = \{(c_j, K_j, s_j)\}_{j=1}^{n}$.

In this work, we aim at making TOD systems better using BLMs. To this end, we propose **S**upervised **In**-**c**ontext TOD (*SincTOD*). Figure 1 shows *SincTOD* in action. For a given test sample (c, K), *SincTOD* predicts a set of hints $\hat{H}$ about the expected response. *SincTOD* then selects exemplars from the training data using $(c, \hat{H})$. Finally, it creates a hint enriched prompt with the exemplars and the test sample and queries a BLM for final response. We now discuss hint prediction, exemplar selection, and prompt creation in details.

3.1 Hint Prediction

For a given (c, K), we consider the following hints about the response s.

1. Entity Types – a list et of types of entities expected in the response s.

2. Dialog Closure – a binary value dc that indicates whether s is the final utterance of the dialog.

3. Response Size – an integer value rs that indicates the number of words in s.

Figure 1 shows the hints for an example dialog. Note that the above hints apply to various domains like restaurant reservations, navigation, hotel booking, etc. Further, assigning hint labels to samples in the training data $\mathcal{D}$ is embarrassingly simple, allowing us to leverage the training data effectively. As hints are unavailable at test time, we learn predictors for them as described below.

Entity Types (ET): For any $(c, K, s) \in \mathcal{D}$, we have list $et = [t_1, t_2, ...]$ as types of the entities present in the response s[1]. We then learn the ET predictor $P(et|c, K)$ on the dataset $\{(c_j, K_j, et_j)\}_{j=1}^{n}$.

Dialog Closure (DC): For any $(c, K, s) \in \mathcal{D}$, we set the label $dc =$ True whenever s is the last utterance in the dialog. Otherwise, we set $dc =$ False. We then learn DC predictor $P(dc|c, K)$ on the dataset $\{(c_j, K_j, dc_j)\}_{j=1}^{n}$.

Response size (RS): For any $(c, K, s) \in \mathcal{D}$, we compute rs as the number of words in the response s. We then learn a RS predictor $P(rs|c, K)$ on the dataset $\{(c_j, K_j, rs_j)\}_{j=1}^{n}$.

We use $H = (et, dc, rs)$ to collectively denote the hints and $\mathcal{D}_h = \{(c, K, s, H)\}_{i=1}^{n}$ to denote hint augmented training data.

3.2 Exemplar Selection

The in-context performance of a BLM depends heavily upon the choice of exemplars (Liu et al., 2021). Further, exemplars semantically closer to the test query often perform better. How can we choose good exemplars for a test sample (c, K) for the TOD task? Intuitively, an exemplar with a dialog state similar to the test's is an ideal choice. However, end-to-end TOD datasets do not include dialog state annotations. Instead, we posit that dialog context and the hints are reasonable proxies for the dialog state. Consequently, in *SincTOD*, we use $(c, \hat{H})$ for exemplar selection.

SincTOD retrieval follows a retrieve-rerank approach (Nogueira and Cho, 2019). First, it dense retrieves the top k samples exemplar store $\mathcal{D}_h$ based on the dialog context. Second, it re-ranks the top k samples by comparing predicted hints $\hat{H}$ with those from the samples. It then selects the top two

[1]We can use a NER tagger to extract these entities, though we assume they are known here.

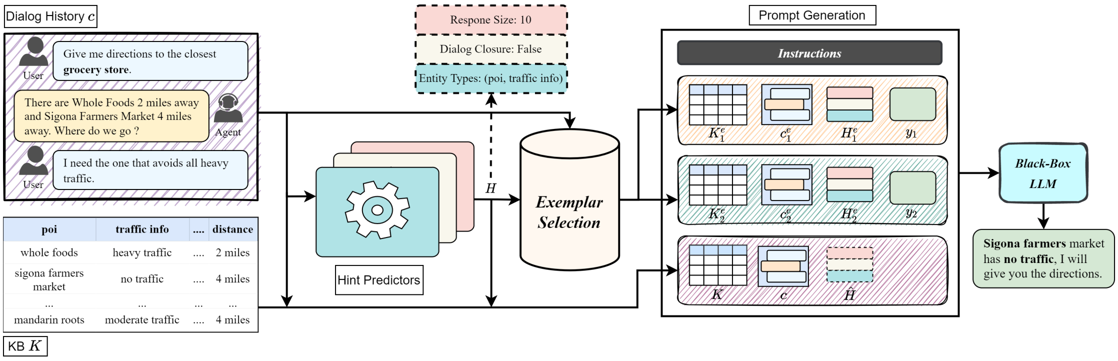

Figure 1: Proposed *SincTOD* model.

samples in re-ranking as the exemplars. We defer to appendix B for further details.

3.3 Prompt Creation

SincTOD prompts comprises of `instructions` followed by tuples `(database, rule, dialog, follow-up response)` for exemplars and test sample.

`instructions` - Task definitions and ontology details for the dataset.

`database` - KB K associated with a sample (exemplar or test). We use JSON index format which we found to perform well during our seed experiments.

`rules` - We include hints H as a set of rules in the prompt and ask the BLM to follow the rules for writing the response. Rules guide the BLM toward the desired answer. We provide further details on rule creation in appendix C.

`dialog history` - User and system utterances in the dialog context c.

`follow-up response` - For exemplars, we succinctly re-iterate the task definition and the entity types expected in the response, followed by gold entities and the response. For the test sample, we only provide task definition and entity types expected in the response and prompt the BLM to generate entities and the final response in order. We refer to this as prompting with **entity generation**. Appendix H shows sample prompts.

4 Experimental Setup

Datasets: We evaluate *SincTOD* on MultiWOZ2.1 (Budzianowski et al., 2018) and Stanford Multi-domain (SMD) (Eric et al., 2017) datasets. More details are given in Appendix A.

Baselines: We compare *SincTOD* against the following baselines - DSR (Wen et al., 2018), KB-Retriever (Qin et al., 2019), GLMP (Wu et al., 2019), DF-Net (Qin et al., 2020), GPT-2+KE (Madotto et al., 2020), EER (He et al., 2020b), FG2Seq (He et al., 2020a), CDNet (Raghu et al., 2021), GraphMemDialog (Wu et al., 2022), ECO (Huang et al., 2022), DialoKG (Rony et al., 2022), UnifiedSKG (Xie et al., 2022), Q-TOD (Tian et al., 2022) and MAKER (Wan et al., 2023). We also report the performance of a vanilla few-shot (*ChatGPT*) prompt. Appendix E provides more details about how the few shots were selected for each input.

We provide the training details for predictor models and retrieval in appendix D.

Model	MultiWOZ		SMD	
	BLEU	Entity F1	BLEU	Entity F1
DSR	9.1	30	12.7	51.9
KB-Retriever	-	-	13.9	53.7
GLMP	6.9	32.4	13.9	60.7
DF-Net	9.4	35.1	14.4	62.7
GPT-2+KE	15.05	39.58	17.35	59.78
EER	13.6	35.6	17.2	59
FG2Seq	14.6	36.5	16.8	61.1
CDNet	11.9	38.7	17.8	62.9
GraphMemDialog	14.9	40.2	18.8	64.5
ECO	12.61	40.87	-	-
DialoKG	12.6	43.5	20	65.9
UnifiedSKG (T5-Large)	13.69	46.04	17.27	65.85
Q-TOD (T5-Large)	17.62	50.61	21.33	71.11
MAKER (T5-large)	**18.77**	54.72	**25.91**	71.3
Few-shot (*ChatGPT*)	8.83	40.25	17.21	70.58
SincTOD (*ChatGPT*)	14.33	52.99	22.08	71.60
SincTOD (*gpt-4*)	13.01	**54.99**	19.08	**72.99**

Table 1: Performance of *SincTOD* and baselines on MultiWOZ and SMD datasets.

5 Results

Table 1 shows the performance of various models on Entity F1 (Wu et al., 2019) and BLEU (Papineni et al., 2002). Across both datasets, the *SincTOD*

Model 1	Model 2	Model 1 Wins	Model 2 Wins	Draws
MAKER	*SincTOD*	5	**25**	30
Gold	*SincTOD*	14	**17**	29
Gold	MAKER	24	11	25

Table 2: Human Evaluation of *SincTOD* (*gpt-4*) on MultiWOZ dataset

	gpt-4	*ChatGPT*
SincTOD	**48.67**	**52.99**
SincTOD w/o Entity Generation	48.28	49.67
SincTOD w/o Hints	37.29	40.25

Table 3: Ablation Study: Entity F1 achieved by *SincTOD* prompt variants.

variants demonstrate competitive Entity F1 scores, with *SincTOD* (*gpt-4*) outperforming all the supervised baseline models. Further, the simpler few-shot variant (*ChatGPT*) displays stronger entity F1 performance on SMD than MultiWOZ. The main reason for this is the nature of the dialogs in the two datasets. SMD contains dialogs that are more templated and consistent, while MultiWOZ has dialogs with diverse linguistic and phrasing variations. Thus SMD performs well with just few-shot examples.

Unlike Entity F1, *SincTOD* variants perform poorly on the BLEU metric. Upon analysis, *SincTOD* responses effectively conveyed essential information from the KB. These responses have meaningful phrasing but reduced lexical overlap with the gold response, thus impacting BLEU scores. We investigate this further in our human evaluation.

Human Evaluation: We conducted a pairwise comparison of models to determine their relative performance. We requested the annotator to consider the responses' groundedness, fluency, and overall satisfactoriness during the evaluation. We select Gold, MAKER[2], and *SincTOD* (*gpt-4*) for human evaluation. Appendix G discusses human evaluations in more detail. Results are reported in table 2. We randomly pick 60 dialog context-response pairs from MultiWOZ dataset for this experiment. First, we observe that annotators clearly prefer *SincTOD* responses over MAKER. Interestingly, annotators also prefer *SincTOD* over Gold responses. This shows that *SincTOD* outputs high-quality responses by leveraging the superior generation capabilities of BLMs.

Ablations: We form two ablation settings. First, we drop the entity generation from the *SincTOD* `follow-up response` in the prompt. Second, we drop the hints from *SincTOD*. Table 3 reports the result for *SincTOD* with *gpt-4* and *ChatGPT*. While we ran the entire test set through *ChatGPT*, we used just 10% of the test set for *gpt-4* due to cost constraints.

[2]We used code and checkpoints released at https://github.com/18907305772/MAKER to get MAKER responses.

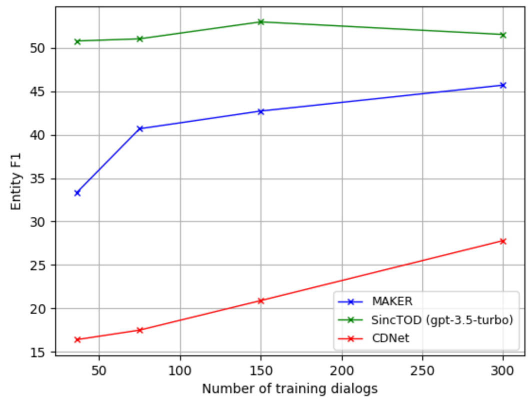

Figure 2: Model performance in low data setting for MultiWOZ dataset.

Low Data Setting: We perform low data experiments with *ChatGPT* due to cost considerations. We evaluate the performance of *SincTOD* (*ChatGPT*) when trained on 36, 75, 150 and 300 dialogs. We adapt *SincTOD* to low data setting as follows. First, we model ET predictor as a multi-label classifier. Then, we learn ET and DC as k-NN classifiers with $k = 10$ and dialog context as neighbor selection. Figure 2 compares the performance of *SincTOD* (*ChatGPT*) with MAKER and CDNet on MultiWOZ dataset. We observe that *SincTOD* (*ChatGPT*) consistently outperforms the baselines in the low data setting.

6 Conclusion

We propose *SincTOD* that leverages BLMs for the end-to-end TOD task. Given a dialog history and KB, *SincTOD* predicts hints about the expected response. It then uses predicted hints for retrieving the exemplars and for guiding a BLM toward desired response. We showed with automatic/human evaluation that *SincTOD* outperforms the SoTA baseline models. Further, *SincTOD* also showcases a strong performance in low-data setting.

Limitations

In our experiments, we work pairs *SincTOD* with commercial black-box LLMs (*ChatGPT* and *gpt-4*). It would be interesting to see if *SincTOD* retains its performance when paired with an open-source LLMs like Llama-2 (Touvron et al., 2023). Further, *SincTOD* is only tested on English dataset though by its design model can easily be extended to different languages. Finally, *SincTOD* performance can further be improved by designing much sophisticated hints.

References

2022. Chatgpt. https://openai.com/blog/chatgpt.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *ArXiv*, abs/2302.04023.

BIG bench authors. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, John A. Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuan-Fang Li, Scott M. Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *ArXiv*, abs/2303.12712.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Conference on Empirical Methods in Natural Language Processing*.

Hyung Won Chung, Le Hou, S. Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Wei Yu, Vincent Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed Huai hsin Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *ArXiv*, abs/2210.11416.

Mihail Eric, Lakshmi. Krishnan, François Charette, and Christopher D. Manning. 2017. Key-value retrieval networks for task-oriented dialogue. *ArXiv*, abs/1705.05414.

Google. 2023. Palm 2 technical report. *ArXiv*, abs/2305.10403.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing.

Zhenhao He, Yuhong He, Qingyao Wu, and Jian Chen. 2020a. Fg2seq: Effectively encoding knowledge for end-to-end task-oriented dialog. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8029–8033.

Zhenhao He, Jiachun Wang, and Jian Chen. 2020b. Task-oriented dialog generation with enhanced entity representation. In *Interspeech*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and L. Sifre. 2022. Training compute-optimal large language models. *ArXiv*, abs/2203.15556.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *ArXiv*, abs/2005.00796.

Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A. Smith, and Mari Ostendorf. 2022. In-context learning for few-shot dialogue state tracking. In *Conference on Empirical Methods in Natural Language Processing*.

Guanhuan Huang, Xiaojun Quan, and Qifan Wang. 2022. Autoregressive entity generation for end-to-end task-oriented dialog. *ArXiv*, abs/2209.08708.

Vojtech Hudecek and Ondrej Dusek. 2023. Are llms all you need for task-oriented dialogue? *ArXiv*, abs/2304.06556.

Zekun Li, Baolin Peng, Pengcheng He, Michel Galley, Jianfeng Gao, and Xi Yan. 2023. Guiding large language models via directional stimulus prompting. *ArXiv*, abs/2302.11520.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? In *Workshop on Knowledge Extraction and Integration for Deep Learning Architectures; Deep Learning Inside Out*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Andrea Madotto, Samuel Cahyawijaya, Genta Indra Winata, Yan Xu, Zihan Liu, Zhaojiang Lin, and Pascale Fung. 2020. Learning knowledge bases with parameters for task-oriented dialogue systems. *ArXiv*, abs/2009.13656.

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. *ArXiv*, abs/1804.08217.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *ArXiv*, abs/1901.04085.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics*.

Xiao Pu, Mingqi Gao, and Xiaojun Wan. 2023. Summarization is (almost) dead. *ArXiv*, abs/2309.09558.

Libo Qin, Yijia Liu, Wanxiang Che, Haoyang Wen, Yangming Li, and Ting Liu. 2019. Entity-consistent end-to-end task-oriented dialogue system with kb retriever. *ArXiv*, abs/1909.06762.

Libo Qin, Xiao Xu, Wanxiang Che, Yue Zhang, and Ting Liu. 2020. Dynamic fusion network for multi-domain end-to-end task-oriented dialog. In *Annual Meeting of the Association for Computational Linguistics*.

Dinesh Raghu, Atishya Jain, Mausam, and Sachindra Joshi. 2021. Constraint based knowledge base distillation in end-to-end task oriented dialogs. *ArXiv*, abs/2109.07396.

Lina Maria Rojas-Barahona, Milica Gašić, Nikola Mrksic, Pei hao Su, Stefan Ultes, Tsung-Hsien Wen, Steve J. Young, and David Vandyke. 2016. A network-based end-to-end trainable task-oriented dialogue system. In *Conference of the European Chapter of the Association for Computational Linguistics*.

Md. Rashad Al Hasan Rony, Ricardo Usbeck, and Jens Lehmann. 2022. Dialokg: Knowledge-structure aware task-oriented dialogue generation. *ArXiv*, abs/2204.09149.

Xin Tian, Yingzhan Lin, Mengfei Song, Siqi Bao, Fan Wang, H. He, Shuqi Sun, and Hua Wu. 2022. Q-tod: A query-driven task-oriented dialogue system. In *Conference on Empirical Methods in Natural Language Processing*.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Fanqi Wan, Weizhou Shen, Ke Yang, Xiaojun Quan, and Wei Bi. 2023. Multi-grained knowledge retrieval for end-to-end task-oriented dialog. In *Annual Meeting of the Association for Computational Linguistics*.

Haoyang Wen, Yijia Liu, Wanxiang Che, Libo Qin, and Ting Liu. 2018. Sequence-to-sequence learning for task-oriented dialogue with dialogue state representation. In *International Conference on Computational Linguistics*.

Chien-Sheng Wu, Richard Socher, and Caiming Xiong. 2019. Global-to-local memory pointer networks for task-oriented dialogue. *ArXiv*, abs/1901.04713.

Jie Wu, Ian G. Harris, and Hongzhi Zhao. 2022. Graphmemdialog: Optimizing end-to-end task-oriented dialog systems using graph memory networks. In *AAAI Conference on Artificial Intelligence*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir R. Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. *ArXiv*, abs/2201.05966.

Steve J. Young, Milica Gasic, Blaise Thomson, and J. Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101:1160–1179.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori Hashimoto. 2023. Benchmarking large language models for news summarization. *ArXiv*, abs/2301.13848.

A Dataset Details

We use the versions of the dataset released by Wan et al. (2023).

Dataset	Domain	#train	#val	#test
MultiWOZ	Restaurant, Hotel, Attraction	1839	117	141
SMD	Navigate, Schedule, Weather	2425	302	304

Table 4: Evaluation Dataset Details

B Exemplar Selection Details

Let $(c, K, \hat{H})$ be the test sample with predicted hints $\hat{H}$. $\mathcal{D}_h = \{(c_j, K_j, s_j, H_j)\}_{j=1}^{n}$ is hint-augmented training data.

Retrieval: We encode each dialog context c_j with a pre-trained language model and for a dense index for points in $\mathcal{D}_h$. Similarly, We encode the test dialog context c and perform a maximum inner-product search (MIPS) to retrieve the top k samples from the augmented training data. For all our experiments we use *BAAI/bge-large-en-v1.5* pre-trained encoder model (Xiao et al., 2023).

Re-ranking: Let H_j be the hints from a retrieved exemplar. We compute similarity score between $\hat{H}$ and H_j as follows

$$f_h(\hat{H}, H_j) = 0.5*\mathbb{1}[\hat{dc} = dc_j] + 0.5*\mathcal{J}(\hat{et}, et_j)$$

where $\mathbb{1}$ is an indicator function and $\mathcal{J}$ is Jaccard similarity. From k retrieved samples, *SincTOD* selects the top two with the highest hint similarity score as exemplars.

C Prompt Creation Details

Creating rules from hints: We transform hints $H = (et, dc, rs)$ to rules in the prompt as follows. For response size, We add a rule `The response must be` rs `words or shorter`. For dialog closure dc = True(False), we add a rule `The response must (not) close the dialog.`. For entity types $et = [t_1, t_2, t_3]$, we add a rule `The response must only include entities of type -` t_1, t_2, t_3`.`. We also introduce a rule `The response must not include any entities of type -` $t'_1, t'_2, ..$ where t' are entity types not present in et. We find that explicitly presenting negative entity types demotivates BLM from including extraneous entities in the response.

D Training Details

We use Nvidia V100 GPUs to train all our models.

ET Predictors: We model all the ET predictors as *flan-t5-large* (Chung et al., 2022) sequence predictors and train them for 8 epochs with a learning rate (LR) of $1e-4$ and batch size (BS) of 32. We use a linear decay LR scheduler with a warm-up ratio of 0.1. We use AdamW optimizer (Loshchilov and Hutter, 2017). Training time was around 10 hours.

DC Predictors: We model all the DC predictors as *deberta-V3-base* (He et al., 2021) binary classifiers and train them for 5 epochs with an LR of $3e-5$, BS of 16, and linear decay LR scheduler with a warm-up ratio of 0.1. We use AdamW optimizer. Training time was around 1 hour.

RS Predictors: During our experiments, we found that training RS predictor is unstable. Thus, we use a constant RS predictor with a value equal to the mean response size in training data.

Exemplar Retrieval: For MultiWOZ dataset, we use the last user utterance in the dialog context to dense retrieve $k = 30$ samples from the training data. We then re-rank them based on the hints and pick top two.

For SMD dataset, we found that retrieval using the entire dialog context works the best. We attribute it to shorted dialog context and utterances in SMD dataset. Further, we use $k = 2$ as exemplars are already of high quality. We use hint re-ranking for deciding the order of the exemplars in the prompt.

E Two shot (*ChatGPT*) Baseline

Let (c, K) be the given test sample. We follow the dense retrieval approach discussed in appendix B and select top two exemplars from the training data. We then prepare prompt as given in section 3 but without the `rules` and entity generation.

For MultiWOZ dataset, we use the last user utterance in the dialog context for the retrieval. For SMD dataset, we use the entire dialog context.

F SMD low data setting results

Figure 3 compares performance of *SincTOD* (*ChatGPT*), MAKER and CDNet on 36, 75, 150 and 300 training dialog from SMD dataset. As in MultiWOZ dataset, *SincTOD* (*ChatGPT*) consistently outperforms the baselines in the low data setting.

G Human Evaluation Details

A snapshot of our human evaluation portal is given in figure 4. Detailed evaluation guidelines are given at the end of this section.

In this work, we human-evaluate responses from three TOD systems - Gold (M_1), MAKER(M_2), and *SincTOD* (*gpt-4*) (M_3). We randomly sample 60 dialog context-response pairs from the MultiWOZ dataset. Two annotators, undergraduate and graduate student volunteers, then independently

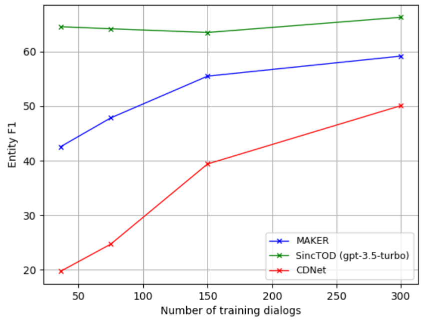

Figure 3: Model performance in low data setting for MultiWOZ dataset.

rank TOD system responses for these 60 samples according to evaluation guidelines.

We then analyze the results for a pair of TOD systems M_1 and $M2$. For a given evaluation sample, we declare M_1 as the winner when a) at least one of the annotators ranks M_1 above M_2, and b)none of the annotators rank M_2 above M_1. Similarly, we declare a draw when the annotators rank M_1 and M_2 the same. Finally, we compute the total number of wins, losses, and draws for M_1 against M_2 and declare the final winner. We report the winners for all (Gold, MAKER), (Gold, *SincTOD* (*gpt-4*)), and (MAKER, *SincTOD* (*gpt-4*)) pairs.

Dear volunteer,

Thank you very much for contributing your valuable time and effort to this task, which is integral to the advancement of conversational systems. This document provides detailed instructions for the annotation task, outlining the specifics on how to annotate the data.

Task Overview

Each data sample has the following key elements:

1. **Dialog History**: A conversation between a user and an assistant, where the assistant helps the user with tasks such as restaurant reservation, hotel booking, or attraction information.

2. **Knowledge Base (KB)**: A database linked to the dialog history.

3. **Responses 1-3**: Three potential continuations to the dialog history.

Annotation Criteria

Your task is to rank the responses 1-3 according to your preference for their suitability as a continuation of the dialogue. You must consider the following criteria for evaluating each response.

1. Groundedness
   - Evaluate if the response is factually accurate given the dialog history and information available in the Knowledge Base (KB).
   - Consider alignment with established context and knowledge within the conversation.

2. Fluency
   - Evaluate the response for grammatical correctness, coherence, and natural language flow.
   - Consider if the response is easily understandable and reads like a human-generated conversation.

3. Satisfaction
   - Assess your overall satisfaction with the response in terms of its appropriateness and effectiveness in addressing the user's needs or queries.
   - Consider the response's completeness, relevance, and general effectiveness in continuing the conversation and fulfilling the user's requirements.

How to Rank?

1. Assign a rank of 1, 2, or 3, where 1 indicates the best and 3 the least favorable response.

2. You can assign the same rank to two or more responses if you find them equally good or bad.

3. Ensure to assign at least one response the rank of 1. Some examples of valid ranking configurations are (1, 2, 3), (1, 2, 2), (1, 1, 2). Some examples of invalid ranking configurations are (2, 2, 3), (3, 2, 3), (3, 3, 3).

Conversation-637

name	food	address	area	phone	postcode	pricerange	type	choice	ref
curry garden	indian	106 regent street city centre	centre	01223302330	cb21dp	expensive	restaurant	both	wc1zy82v
the missing sock	international	finders corner newmarket road	east	01223812660	cb259aq	cheap	restaurant	both	wc1zy82v
pizza hut city centre	italian	regent street city centre	centre	01223323737	cb21ab	cheap	restaurant	both	wc1zy82v
bloomsbury restaurant	international	crowne plaza hotel 20 downing street	centre	08719429180	cb23dt	moderate	restaurant	both	wc1zy82v
the varsity restaurant	international	35 saint andrews street city centre	centre	01223356060	cb23ar	moderate	restaurant	both	wc1zy82v

1 what restaurants in the centre serve international cuisine ?

2 the varsity restaurant and the bloomsbury restaurant serve international food and are in the centre of town .

3 how about a place in the moderate price range ?

4 both of the named restaurants are in the moderate price range .

5 ok , can you book a table for 6 at 12:00 on tuesday at the varsity restaurant ? i will need a reference number too , please .

Response - 1 "i ' m sorry , but there are no tables available at that time . would you like to try another restaurant ?"

Response - 2 "certainly . i will have that reference number for you in just one second ."

Response - 3 "i ' m sorry , but i can ' t provide the booking information you ' re asking for ."

Rank the above responses based on your preference for their suitability as a continuation of the dialogue. You must consider the groundedness, fluency and satisfaction criteria when you evaluate the responses.

Response-1	Response-2	Response-3
--select rank--	--select rank--	--select rank--

Figure 4: Portal

H Prompts

MultiWOZ

```
Henceforth, assume that you are a customer support expert. I will give you an incomplete dialog between a user and a customer service representative. As an expert, you must suggest the most appropriate follow-up response to the dialog. Ensure you also include correct information (entities) from the given database. Entities can be of the following types -
1. name - name of a place (restaurant, hotel or attraction)
2. address - address of the place
3. phone - phone number of the place
4. food - the type of food a restaurant serves
5. area - a region of the city, e.g. centre, north, south, east, west
6. postcode - postcode of the place
7. price range - price range of the place, e.g. cheap, moderate, expensive
8. type - the type of a place, e.g. restaurant, hotel, gusesthouse, attraction
9. reference number - reference code for booking, e.g. 542j9wog
10. stars - star rating of the hotel, e.g. 3 stars
11. choice - number of available choices that match user's requirements, e.g. many, few, several, 10

As an expert, you are very strict about following rules. Make sure that the follow-up response you write follows all the given rules. Here are the examples -

[example 1]
[database 1]
{
    "magdalene college":{
        "address":"magdalene street",
        "phone":"01223332138",
        "area":"west",
        "postcode":"cb30ag",
        "price range":"free",
        "type":"college",
        "choice":"79"
    },....
}

[rules 1]
The response must be 15 words or shorter.
The response must not close the dialog.
The response must only include entities of type - choice.
The response must not include any entities of type - name, address, phone, food, area, postcode, price range, type, reference number, stars.

[dialog history 1]
user: hello i am looking for a place to go , can you help me ?

[follow-up response 1]
Let's think step-by-step.
As an expert, I must understand the user's requirements from [dialog history 1], identify the relevant information from the [database 1], follow all the [rules 1] and write the response.
I will include entities of type ['choice'] in my response.
I will include these entities - [('choice', '79')]
assistant: i have found 79 place for you . do you have any specific ideas in mind ?

[example 2]
[database 2]
{
    äcorn guest house":{
        "address":"154 chesterton road",
        "phone":"01223353888",
        "area":"north",
        "postcode":"cb41da",
        "price range":"moderate",
        "type":"guesthouse",
        "stars":"4 star",
        "choice":"24"
    },....
}

[rules 2]
The response must be 10 words or shorter.
The response must not close the dialog.
The response must only include entities of type - choice.
The response must not include any entities of type - name, address, phone, food, area, postcode, price range, type, reference number, stars.

[dialog history 2]
user: i ' d like to find a guesthouse to stay .

[follow-up response 2]
Let's think step-by-step.
As an expert, I must understand the user's requirements from [dialog history 2], identify the relevant information from the [database 2], follow all the [rules 2] and write the response.
I will include entities of type ['choice'] in my response.
I will include these entities - [('choice', '24')]
assistant: no problem . we have 24 to choose from . any specifics ?

[example 3]
[database 3]
{
    "great saint mary ' s church":{
        "address":"market square",
        "phone":"01223350914",
        "area":"centre",
        "postcode":"cb23pq",
        "price range":"cheap",
        "type":"architecture",
        "choice":"a lot"
    },....
}

[rules 3]
The response must be 15 words or shorter.
The response must not close the dialog.
The response must only include entities of type - choice.
The response must not include any entities of type - name, address, phone, food, area, postcode, price range, type, reference number, stars.

[dialog history 3]
user: i am looking for a place to go !

[follow-up response 3]
Let's think step-by-step.
As an expert, I must understand the user's requirements from [dialog history 3], identify the relevant information from the [database 3], follow all the [rules 3] and write the response.
I will include entities of type ['choice'] in my response.
I will include these entities -
```

SMD

Henceforth, assume that you are an expert in in-car infotainment. I will give you an incomplete dialog between a user and an in-car infotainment system. As an expert, you must suggest the most appropriate follow-up response to the dialog. Ensure you also include correct information (entities) from the given database. Entities can be of the following types -
1. poi - name of a point of interest, e.g., home, starbucks, pizza chicago, etc.
2. address - address of a poi, e.g, 783 arcadia pl.
3. poi type - the type of a poi, e.g., tea or coffee place, hospital, shopping center, etc.
4. traffic info - traffic status on the way to a poi, e.g., heavy traffic, no traffic, road block nearby, etc.
5. distance - distance of a poi from the user's current location, e.g., 2 miles, 4 miles, etc.
6. event - an event in the user's calendar
7. date - date in a month like the 1st or the 4th or day of a week like monday, wednesday.
8. time - the time on which an event is scheduled
9. party - party attending an event, e.g., tom, boss, brother, executive team, etc.
10. agenda - agenda associated with an event, e.g., discuss dress code, go over budget, etc.
11. room - meeting place of an event, e.g., conference room 100, etc.
12. location - a location for which the user may request the weather information, e.g, boston, los angeles, etc.
13. weather attribute - weather description in a location, e.g., cloudy, warm, hot, overcast etc.
14. temperature - the in a location, e.g., 60f, 100f, etc.
15. weekly time - temporal indicators like today, tomorrow, next week etc.

As an expert, you are very strict about following rules. Make sure that the follow-up response you write follows all the given rules. Here are the examples -

[example 1]
[database 1]

```
{
    "trader joes":{
        "address":"408 university ave",
        "poi type":"grocery store",
        "traffic info":"no traffic",
        "distance":"5 miles"
    },....
}
```

[rules 1]
The response must be 11 words or shorter.
The response must not close the dialog.
The response must only include entities of type - poi, poi type.
The response must not include any entities of type - address, traffic info, distance, event, date, time, party, agenda, room, location, weather attribute, temperature, weekly time.

[dialog history 1]
user: give me directions to the nearest grocery store

[follow-up response 1]
Let's think step-by-step.
As an expert, I must understand the user's requirements from [dialog history 1], identify the relevant information from the [database 1], follow all the [rules 1] and write the response.
I will include entities of type ['poi', 'poi type'] in my response.
I will include these entities - [('poi', 'trader joes'), ('poi type', 'grocery store')]
system: the nearest grocery store is trader joes , would you like directions ?

[example 2]
[database 2]

```
{
    "safeway":{
        "address":"452 arcadia pl",
        "poi type":"grocery store",
        "traffic info":"heavy traffic",
        "distance":"4 miles"
    },....
}
```

[rules 2]
The response must be 23 words or shorter.
The response must not close the dialog.
The response must only include entities of type - distance, poi, traffic info.
The response must not include any entities of type - address, poi type, event, date, time, party, agenda, room, location, weather attribute, temperature, weekly time.

[dialog history 2]
user: give me directions to the closest grocery store

[follow-up response 2]
Let's think step-by-step.
As an expert, I must understand the user's requirements from [dialog history 2], identify the relevant information from the [database 2], follow all the [rules 2] and write the response.
I will include entities of type ['distance', 'poi', 'traffic info'] in my response.
I will include these entities - [('distance', '4 miles'), ('poi', 'safeway'), ('traffic info', 'heavy traffic')]
system: we ' re 4 miles away from safeway but there is heavy traffic in this moment : do i set the gps to go there ?

[example 3]
[database 3]

```
{
    "sigona farmers market":{
        "address":"638 amherst st",
        "poi type":"grocery store",
        "traffic info":"no traffic",
        "distance":"4 miles"
    },....
}
```

[rules 3]
The response must be 10 words or shorter.
The response must not close the dialog.
The response must only include entities of type - distance, poi, poi type.
The response must not include any entities of type - address, traffic info, event, date, time, party, agenda, room, location, weather attribute, temperature, weekly time.

[dialog history 3]
user: give me directions to the closest grocery store

[follow-up response 3]
Let's think step-by-step.
As an expert, I must understand the user's requirements from [dialog history 3], identify the relevant information from the [database 3], follow all the [rules 3] and write the response.
I will include entities of type ['distance', 'poi', 'poi type'] in my response.
I will include these entities -