Synergizing In-context Learning with Hints for End-to-end Task-oriented Dialog Systems

Anonymous ACL submission

Abstract

This work explores the effectiveness of large language models (LLMs) for end-to-end task-oriented dialog systems. We evaluate *Llama2*, *ChatGPT*, and *gpt-4* in the few-shot (in-context) setting on two end-to-end TOD datasets and find that their performance is not on par with the existing SoTA models. We posit that, unlike the SoTA models, LLM responses do not align well with the training data due to their limited context size. In response, we propose *SyncTOD*, which synergizes LLMs with useful hints about the task for improved alignment. At a high level, *SyncTOD* uses the auxiliary models to provide these hints and exemplar selection for the in-context prompts. With *gpt-4*, *SyncTOD* outperforms SoTA models on MultiWOZ and SMD datasets. Further, *SyncTOD* achieves superior performance compared to LLMs and SoTA models in low-data settings while retaining competitive performance in full-data settings.

1 Introduction

The rise of large-language models (LLMs) has progressed the field of NLP by leaps and bounds (Google, 2023; Touvron et al., 2023). Pre-trained over massive data, LLMs work remarkably well with just in-context learning for many NLP tasks like natural language inference, summarization, and dialogs (Kavumba et al., 2023; Hu et al., 2022; Zheng et al., 2023).

Task-oriented dialog (TOD) systems generate agent response, given the dialog history and an associated task-specific knowledge base (KB). There are two flavours of TOD systems: modular (Young et al., 2013; Rojas-Barahona et al., 2016) and end-to-end (Madotto et al., 2018; Wu et al., 2019). Modular systems require domain experts to define dialog states for the domain and annotate each train dialogs with these dialog state annotations. Unlike modular, end-to-end systems do not require any such expensive state annotations. Our focus is to study in-context learning of end-to-end TOD systems.

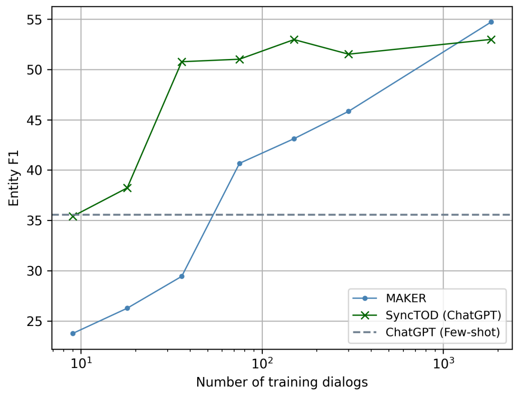

Figure 1: Model performance on MultiWOZ dataset at different training data sizes.

End-to-end systems built using in-context learning performs better than supervised approaches only when the number of dialogs available for training is small. For instance, Figure 1 shows the entity F1 achieved by few-shot *ChatGPT* against MAKER (Wan et al., 2023), a SoTA supervised approach, on the MultiWOZ dataset (Budzianowski et al., 2018). While LLMs can adapt quickly to new tasks with minimal examples, somewhat unexpectedly, MAKER outperformed *ChatGPT* when provided with just 50 examples in a multi-domain dataset such as MultiWOZ.

To understand this behaviour, we use sample responses generated by various models in Table 1. We see that *ChatGPT*'s response lacks alignment with the nature of the response expected from the system. MAKER (and gold) often seek additional user input when posed with an excessive number of options. In contrast, *ChatGPT* tends to respond with a comprehensive answer, listing all potential options. This verbosity hinders the user's ability to

name	address	...	price range
Arbury Lodge Guesthouse	82 Arbury road	...	moderate
Home from Home	78-80 Milton road	...	moderate
Hamilton Lodge	156 Chesterton road	...	moderate
Worth House	152 Chesterton road	...	cheap
Acorn Guest House	154 Chesterton road	...	moderate
City Centre North B and B	328a Histon road	...	cheap

User	I am looking for a place to stay. The hotel should include free wifi and should include free parking.
Gold	There are many results with free wifi and parking, to help narrow it down, what area do you want to stay in?
MAKER	There are many hotels that offer both free wifi and parking. Do you have a preference on the area of town?
ChatGPT (Few-shot)	There are several guesthouses in the north area that you can consider. Some options include Arbury Lodge Guesthouse, ..., and City Centre North B&B. These guesthouses offer different star ratings and price ranges. However, I'm not sure if they provide free wifi and parking. Would you like me to check that for you?
SyncTOD (ChatGPT)	We have many options available. Do you have a specific area in mind?

Table 1: *ChatGPT* response is misaligned with the training data. The response is verbose and .

grasp the information easily.

To alleviate this issue and improve the performance of in-context learning, we propose Synergizing In-context with hints for TOD (*SyncTOD*) to align the responses generated by LLMs to the gold responses. We show that when these hints are used to dynamically pick suitable exemplars can further improve the generated responses.

On two publicly available datasets, *SyncTOD* outperforms both vanilla prompting and SoTA supervised models in low-data setting while maintaining a competitive performance in the full-data setting. With a mere 36 examples, the entity F1 of *SyncTOD* is comparable to MAKER trained with approximately 1000 examples, and 10-15 points more than vanilla prompting of LLMs.

2 *SyncTOD*

Let $c = [u_1, a_1, u_2, a_2, ..., u_j]$ be a user-agent dialog history with u and a being user and agent utterances respectively. Let $y = a_j$ be the next agent response. Let K be the KB associated with the user's task. The aim is to predict the next agent response $\hat{y}$ given the dialog history c and the associated KB K.

In an in-context learning setup, the next agent response $\hat{y}$ is predicted with a prompt containing task instructions, exemplars, and (c, K). We consider the setting where a training set $\mathcal{D} = \{(c_i, K_i, y_i)\}_{i=1}^{n}$ for the task is available.

Our proposed approach, *SyncTOD*, synergizes in-context learning with hints to better align with responses in training data. Our approach has two main parts. (1) *Hint Prediction* module that predicts hints necessary to guide the LLM. These hints predictors are learnt using the training set $\mathcal{D}$. (2) *Exemplar Retrieval* module that uses the predicted hints to select exemplars from $\mathcal{D}$ via a retrieve-rerank strategy. We now discuss both these modules in detail and defer to appendix G for the exact prompt design.

2.1 Hint Prediction

We propose to use three types of hints which includes entity types (in response), response length, and dialog closure. Our choice of hints has two advantages. First, they are domain agnostic. Second, as discussed below, the hint prediction models can be learnt using distant supervision, without the need for any manual annotations.

Entity Types (ET): For a $(c, K, y) \in \mathcal{D}$, we define entity types $et = [t_1, t_2, ...]$ as the list of entity types present in response y. Following prior works, we find et by simply matching entities from K in the response y (Wu et al., 2019; Raghu et al., 2021). Finally, we learn a ET predictor $P(et|c, K)$ on the dataset $\{(c_i, K_i, et_i)\}_{i=1}^{n}$.

Response size (RS): For a $(c, K, y) \in \mathcal{D}$, we define response size rs equal to number of words in the response y. We learn an RS predictor $P(rs|c, K)$ on the dataset $\{(c_i, K_i, rs_i)\}_{i=1}^{n}$.

Dialog Closure (DC): For a $(c, K, y) \in \mathcal{D}$, we define dialog closer dc = True if and only if y is the last utterance in the dialog. We then learn a DC predictor $P(dc|c, K)$ on the dataset $\{(c_i, K_i, dc_i)\}_{i=1}^{n}$.

For a test dialog (c, K), *SyncTOD* predicts the hints $\hat{H} = (\hat{et}, \hat{rs}, \hat{dc})$ using ET, RS and DC hint predictors respectively.

2.2 Exemplar Retrieval

SyncTOD has a retrieve-rerank mechanism for selecting in-context exemplars (Nogueira and Cho, 2019). Following (Liu et al., 2021), *SyncTOD* selects points from $\mathcal{D}$ that are semantically closer to the given test dialog (c, K). Specifically, it encodes the dialog context c using a pre-trained encoder and performs a maximum inner-product search (MIPS) over $\mathcal{D}$ to retrieve the top-k points. In all our experiments, we use *BAAI/bge-large-en-v1.5* pre-trained

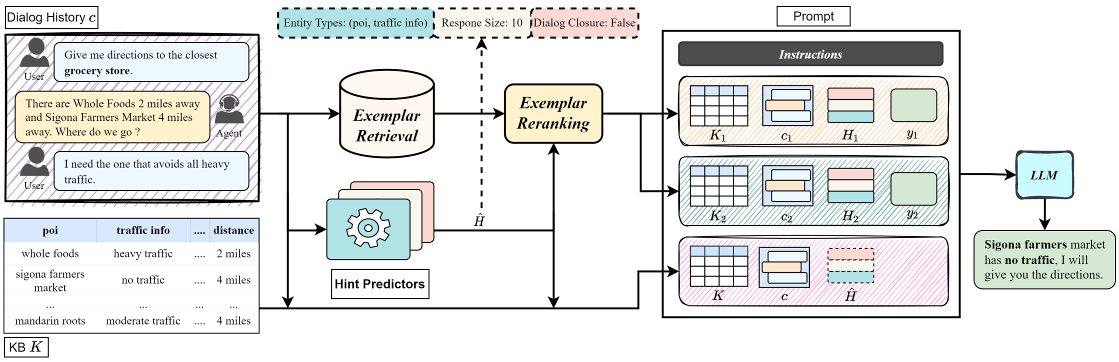

Figure 2: Proposed *SyncTOD* model.

encoder model (Xiao et al., 2023).

2.3 Exemplar Re-ranking

Intuitively, an example with the same dialog state as the input is an ideal choice for an exemplar. However, end-to-end TOD datasets do not include dialog state annotations. Instead, we posit that dialog context along with the hints are reasonable proxies for the dialog state. *SyncTOD* thus re-ranks the retrieved points based on hints.

Let (c_i, K_i, y_i) be a retrieved point and H_is be its associated hints. *SyncTOD* computes similarity score between hints $\hat{H}$ and H_i as follows

$$f_h(\hat{H}, H_i) = 0.5 * \mathbb{1}[\hat{dc} = dc_i] + 0.5 * \mathcal{J}(\hat{et}, et_i)$$

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

3 Experimental Setup

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

Baselines: We compare *SyncTOD* against the following recent baselines - 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 *Llama2*, *ChatGPT* in a standard few-shot setting with fixed exemplars. We set the decoding temperature to zero for all the LLMs in our experiments to obtain reproducible responses. We defer comparison with additional baselines and *Llama2* chat models in appendix C.

Model	MultiWOZ		SMD	
	BLEU	Entity F1	BLEU	Entity F1
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
ChatGPT (zero-shot)	3.39	28.16	6.91	60.11
ChatGPT (few-shot)	8.83	40.25	17.21	70.58
Llama2 70B (few-shot)	5.26	39.68	3.29	46.20
SyncTOD (*Llama2* 70B)	14.44	50.51	15.37	63.33
SyncTOD (*ChatGPT*)	14.33	52.99	22.08	71.60
SyncTOD (*gpt-4*)	13.01	**54.99**	19.08	**72.99**

Table 2: Performance of *SyncTOD* and baselines on MultiWOZ and SMD datasets.

4 Results

Full-data setting: Table 2 shows the performance of various models on Entity F1 (Wu et al., 2019) and BLEU (Papineni et al., 2002). We provide the training details for *SyncTOD* hint predictors and retrieval in Appendix D.

Across both datasets, vanilla few-shot LLMs perform terribly compared to the baselines, whereas *SyncTOD* variants demonstrate competitive Entity F1 scores, with *SyncTOD* (*gpt-4*) outperforming all the supervised baseline models. Importantly, both *Llama2* and *ChatGPT* LLMs enjoy consistent performance gains when coupled with *SyncTOD*. 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 more templated and consistent dialogs, while MultiWOZ has dialogs with diverse linguistic and phrasing variations. Thus, SMD performs well with just a few examples.

Unlike Entity F1, *SyncTOD* variants perform

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

Table 3: Human Evaluation of *SyncTOD* (*gpt-4*) on MultiWOZ dataset

Accuracy	MultiWOZ	SMD
Closure Prediction	0.9564	0.9109
Entity Type Prediction	0.6805	0.7436

Table 4: Accuracy of hint Predictor models.

	MultiWOZ	SMD
SyncTOD (*ChatGPT*)	52.99	71.60
w\o hint prediction	40.60	70.77
w\o exemplar retrieval	45.47	66.84
w\o exemplar reranking	49.94	71.60

Table 5: Ablation Study: Entity F1 on MultiWOZ and SMD datasets

poorly on the BLEU metric. Upon analysis, *SyncTOD* 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 conduct human evaluation over MultiWOZ dataset with Gold, MAKER[1], and *SyncTOD* (*gpt-4*). Appendix F discusses human evaluations in greater detail.

On a high level, we task an annotator to evaluate responses from two models based on groundedness, fluency, and overall satisfactoriness. Post evaluation, the annotator can show his/her preference for one of the two responses. We then declare the model preferred by the annotator as the winner. We repeat this exercise with two annotators and 60 randomly picked dialog histories from the test set and report the aggregate results in table 3. We find that annotators clearly prefer *SyncTOD* responses over MAKER. Interestingly, annotators also prefer *SyncTOD* over Gold responses. This shows that *SyncTOD* outputs high-quality responses by leveraging the superior generation capabilities of LLMs.

Hint Predictors Performance: Table 4 reports the performance of *SyncTOD* hint predictors. We report accuracy for DC predictor and micro F1 for ET predictor. We compute micro F1 for ET predictor as follows: Let sets G and P be gold and predicted entity types for a given response, calculate true positives $TP = |G \cap P|$, false positives $FP = |P/G|$ and false negatives $FN = |G/P|$. Then, use TP, FP, and FN to compute micro precision, recall, and F1 over the test set.

We observe that the DC predictor achieves high performance across datasets. However, ET predictors still show room for improvement, which indicates *SyncTOD* performance can be pushed further.

Ablations: We perform ablations on *SyncTOD* (*ChatGPT*) and report results in 5. We find hints and exemplar retrieval critical for *SyncTOD* performance across datasets. However, dropping exemplar re-ranking affects MultiWOZ much more than SMD. We attribute this to templated nature of dialogs in SMD that allows *SyncTOD* to retrieve high-quality exemplars without re-ranking.

Low Data Setting: Figure 1 and 3 showcase performance of *SyncTOD* (*ChatGPT*) on MultiWOZ and SMD datasets respectively at increasing training data sizes. To cope with data scarcity, we model *SyncTOD* hint predictors as simple k-nn models where nearest neighbors are selected from the available data using *BAAI/bge-large-en-v1.5* pre-trained encoder model. With limited data, *SyncTOD* consistently boosts *ChatGPT* performance and considerably outperforms MAKER.

5 Conclusion

We propose *SyncTOD* that leverages LLMs for the end-to-end TOD task. Given a dialog history and KB, *SyncTOD* obtains hints about the expected response using auxiliary models. It then uses predicted hints to retrieve quality exemplars and guide LLMs toward the desired response. With automatic/human evaluation, we showed that *SyncTOD* outperforms the SoTA baseline models. Further, *SyncTOD* also showcases a strong performance in the low-data setting.

Limitations

It would be interesting to see how *SyncTOD* benefits from advanced prompting techniques like chain-of-thought and self-consistency. Further, *SyncTOD* is only tested on English datasets, though the model can easily be extended to different languages by its design. Finally, *SyncTOD* performance can further

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

be improved by designing much more sophisticated hints.

References

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.

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 pre-training 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*.

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.

Pride Kavumba, Ana Brassard, Benjamin Heinzerling, and Kentaro Inui. 2023. Prompting for explanations improves adversarial nli. is this true? {Yes} it is {true} because {it weakens superficial cues}. In *Findings*.

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*.

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, Wenbo Pan, Qiguang Chen, Lizi Liao, Zhou Yu, Yue Zhang, Wanxiang Che, and Min Li. 2023. End-to-end task-oriented dialogue: A survey of tasks, methods, and future directions.

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.

Lianmin Zheng, Ying Sheng, Wei-Lin Chiang, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Chatbot arena: Benchmarking llms in the wild with elo ratings. `https://lmsys.org/blog/2023-05-03-arena/`.

A Related Works

Conventional TOD systems follow a modular design (Young et al., 2013; Rojas-Barahona et al., 2016; Hosseini-Asl et al., 2020; Qin et al., 2023) and require annotations for DST, PL and NLG. This work, however, focuses on end-to-end TOD systems (Eric et al., 2017; Madotto et al., 2018; Wu et al., 2019; Qin et al., 2023) that alleviate the need for annotations by directly predicting the response given dialog history and knowledge base (KB).

Though LLMs 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, *SyncTOD* uses multiple hints – entity types, response length, and dialog closure – relevant to the TOD task. Further, *SyncTOD* also uses these hints to improve the in-context exemplars' quality.

B Dataset Details

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

C Additional Baselines

We compared our model against the following end-to-end TOD baselines - We compare *SyncTOD*

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

Table 6: Evaluation Dataset Details

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). Results are shown in table 7.

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
ChatGPT (zero-shot)	3.39	28.16	6.91	60.11
ChatGPT (few-shot)	8.83	40.25	17.21	70.58
Llama2 70B (few-shot)	5.26	39.68	3.29	46.20
Llama2 Chat 70B (few-shot)	3.34	30.33	3.15	53.27
SyncTOD (*Llama2* 70B)	14.44	50.51	15.37	63.33
SyncTOD (*Llama2* Chat 70B)	8.35	48.01	7.92	63.31
SyncTOD (*ChatGPT*)	14.33	52.99	22.08	71.60
SyncTOD (*gpt-4*)	13.01	**54.99**	19.08	**72.99**

Table 7: Performance of *SyncTOD* and baselines on MultiWOZ and SMD datasets.

D Training *SyncTOD* with Full Training Set

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 the 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 the 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 the top two.

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

E SMD low data setting results

Figure 3 compares the performance of *SyncTOD* (*ChatGPT*), and MAKER on an increasing number of training dialogs from SMD dataset. As in MultiWOZ dataset, *SyncTOD* (*ChatGPT*) with simple k-nn predictors consistently outperforms the baselines in the low data setting.

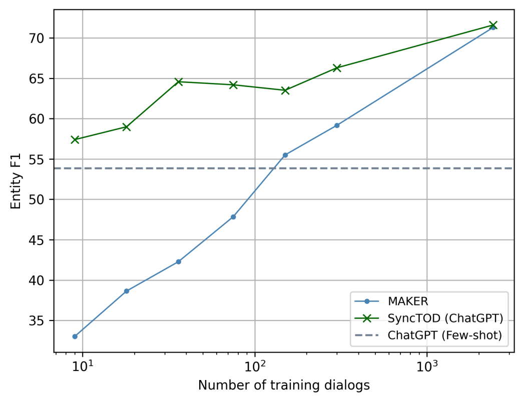

Figure 3: Model performance in low data setting for SMD dataset.

F 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 *SyncTOD* (*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 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, *SyncTOD* (*gpt-4*)), and (MAKER, *SyncTOD* (*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).

G Prompt Specification

G.1 Design

SyncTOD prompts are comprised 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 LLM to follow the rules for writing the response. Rules guide the LLM toward the desired answer. We provide further details on rule creation at the end of this section.

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

`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 LLM to generate entities and the final response in order.

G.2 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 LLM from including extraneous entities in the response.

G.3 Sample 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 -