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

Anonymous ACL submission

Abstract

Large language models (LLM) based end-to-end task-oriented dialog (TOD) systems built using few-shot (in-context) learning perform better than supervised models only when the train data is limited. This is due to the inherent ability of LLMs to learn any task with just a few demonstrations. As the number of train dialogs increases, supervised SoTA models surpass in-context learning LLMs as they learn to better align with the style of the system responses in the training data, which LLMs struggle to mimic. In response, we propose *SyncTOD*, which synergizes LLMs with useful hints about the task for improved alignment. At a high level, *SyncTOD* trains auxiliary models to provide these hints and select exemplars for the in-context prompts. With *ChatGPT*, *SyncTOD* achieves superior performance compared to LLM-based baselines and SoTA models in low-data settings, while retaining competitive performance in full-data settings.

Name	Food	...	Area	Pricerange
Graffiti	British	...	west	expensive
The Oak Bistro	British	...	centre	moderate
The Copper Kettle	British	...	centre	moderate
Travellers Rest	British	...	west	expensive
Saint Johns Chop House	British	...	west	moderate
Restaurant One Seven	British	...	centre	moderate

Context	**User:** Are there any restaurants that serve proper British food in town?
Gold	Oh yes quite a few. Which part of town will you be dining in?
MAKER	There are no British restaurants. Would you like to try a different cuisine?
Few-shot (*GPT-4*)	Yes, there are quite a few restaurants that serve British food. One of them is The Oak Bistro located at 6 Lensfield Road, with the postcode cb21eg. Another option is The Copper Kettle at 4 Kings Parade City Centre, postcode cb21sj. Would you like more options?
SyncTOD (*GPT-4*)	Yes, there are quite a few. Do you have a preference for the price range or area?

Table 1: *GPT-4* lists many potential options and extraneous details instead of seeking user input and lacks alignment with the gold.

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).

One specific domain within dialogs where LLMs show promise is in building Task-Oriented Dialogs (TOD) systems, where they generate agent responses based on the dialog history and task-specific knowledge. TOD systems, in general, can be divided into two types: modular (Young et al., 2013) and end-to-end (Madotto et al., 2018). Modular systems require domain experts to define dialog states and annotate each train dialog with state annotations. Unlike modular, end-to-end systems do not require such expensive state annotations. In this work, we focus on end-to-end TOD systems.

Given a corpus of train dialogs, an ideal end-to-end system should not only learn the underlying task (e.g., restaurant recommendation) but also align with the language and style of the agent utterances in the train dialogs. This alignment is crucial, particularly in scenarios like in-car voice assistants, where responses must be concise and easily consumable without causing distraction.

LLM-based systems built using in-context learning perform better than supervised models when the training dataset is limited. The inherent reasoning capabilities of LLMs help them to learn the associated task with just a few examples. However, they do not align well with the language and style in the train dialogs. Moreover, supervised approaches are better than in-context approaches when a reasonable number of train dialogs are available. These approaches better align with the training set, but are weaker in inherent reasoning ability.

As an illustrative example, see the responses generated by various models in Table 1. We see that *GPT-4* is good at reasoning but lacks alignment in the way in which information is presented. In situations where the gold seeks additional user input when posed with excessive options, *GPT-4* tends to be overly comprehensive, listing many potential options and extraneous details. This verbosity, while informative, can hinder users in easily grasping the information; whereas, MAKER (Wan et al., 2023), a SoTA supervised approach, is well aligned with agent utterances in training, but makes many mistakes in reasoning.

Contributions: We propose *Synergizing in-context learning with hints for TOD (SyncTOD)*, which combines LLM's reasoning with supervised models' task alignment. In particular, it trains auxiliary models, which provide LLMs (accessed via an API) with hints (such as expected entity types in the response and response length) on how to phrase the response; selecting exemplars conditioned on these hints further improves the alignment of the responses. On three publicly available datasets, *SyncTOD* consistently outperforms both vanilla prompting and SoTA supervised models in a limited-data setting while maintaining a competitive performance compared to supervised models in the full-data setting.

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 system response. The task of a TOD system is to predict the next system response $\hat{y}$ given the dialog history c and a knowledge base (KB) K associated with the user's task. Let $\mathcal{D} = \{(h_i, K_i, y_i)\}_{i=1}^n$ denote the train dialogs.

In the in-context learning setup, an LLM is queried (via API) with an input prompt containing task instructions, a few exemplars, and (c, K) to generate $\hat{y}$. A popular technique for leveraging train dialogs in the in-context learning setup is retrieval augmented generation (RAG) (Zhang et al., 2023a; Guu et al., 2020). In RAG, the exemplars that are most similar to c are retrieved from $\mathcal{D}$ and are used for generating $\hat{y}$.

Our proposed approach, *SyncTOD*, synergizes in-context learning of LLMs with *hints* to better align with agent utterances in the training data $\mathcal{D}$. Figure 1 shows the overall architecture. *SyncTOD* has two main components: hint predictors and exemplar selector. The hint predictors output a set of hints $\hat{H}$ given the dialog history c. These hints are domain-agnostic clues, such as the entity types that should be included in the response and the length of the response, that can guide the generation to follow the same style as the train dialogs. The second component, exemplar selector, first retrieves relevant exemplars from $\mathcal{D}$ based on c, and then re-ranks the retrieved exemplars based on $\hat{H}$. Both these components are aimed at aligning the language and style of LLM responses to agent responses in the train dialogs $\mathcal{D}$. As the gold responses y are available for the exemplars, we simply infer the corresponding hints from y and add the hints to the exemplars. The predictors are only used to infer hints for the given input dialog with history c. Please refer to appendix J for the exact prompt.

2.1 Hint Predictors

SyncTOD uses three types of hints: entity types (in response), response length, and dialog closure.

Entity Types (ET): Entities are the information-rich elements in the agent's response. For example, the *hotel* name "Lovell Lodge" is the crucial element in the agent response "How does the Lovell Lodge sound?". We posit that for a given dialog context and KB, the set of entity types in the agent response (e.g., {*hotel name*}) captures the crux of the response. Hence using expected entity types in the response as hints would align the LLM generation to $\mathcal{D}$.

Specifically, for given (c, K), *SyncTOD* predicts a list of entity types $\hat{et}$ present in the expected system response. Then, *SyncTOD* amends the prompt with the rule – *The response must only include entities of type:* $\hat{et}$. To predict $\hat{et}$, *SyncTOD* learns an ET predictor model $P(et|c, K)$ on the dataset $\{(c_i, K_i, et_i)\}_{i=1}^n$, where gold et_is are the types of entities in gold response.

Dialog Closure (DC): The style of the dialog closures varies depending on the task at hand, and each dataset has a different way of closing the dialog. But *ChatGPT* generates similar, verbose and open-ended responses to the user's closing salutations. To alleviate this, *SyncTOD* uses dialog closure prediction dc for a given dialog (c, K) as a hint to steer LLM towards a successful closure

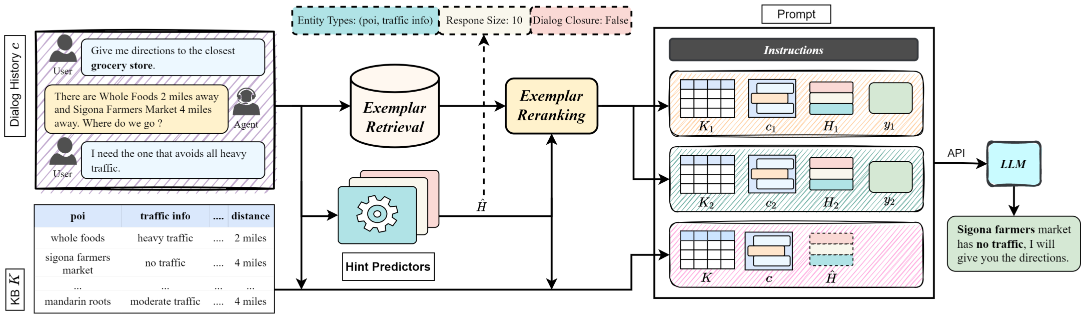


Figure 1: *SyncTOD* predicts useful hints $\hat{H}$ about the expected response. The hints improve exemplar quality via re-ranking and steer the LLM (accessed via API) toward the expected response from within the prompt.

of the dialog. Specifically, *SyncTOD* amends the input prompt with a rule: *The response must close the dialog.*, when dc is true. For a training dialog (c_i, K_i, y_i), we define dc = True if and only if y_i is the last utterance in the dialog.

Response size (RS): For a $(c_i, K_i, y_i) \in \mathcal{D}$, response size rs equals the number of words in the response y_i. *SyncTOD* learns an RS predictor $P(rs|c, K)$ on the dataset $\{(c_i, K_i, rs_i)\}_{i=1}^{n}$ and amends the input with rule: *The response must be rs words or shorter.*

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 Selector

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 history 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* encoder model (Xiao et al., 2023).

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 history and hints are reasonable proxies for the dialog state. *SyncTOD* thus re-ranks the retrieved datapoints based on hints.

Let (c_i, K_i, y_i) be a retrieved datapoint 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), Stanford Multi-domain (SMD) (Eric et al., 2017) and BiTOD (English) (Lin et al., 2021) datasets. More details are given in Appendix B.

Baselines: We compare *SyncTOD* against the 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 compare against RAG with *BAAI/bge-large-en-v1.5* model for exemplar retriever.

We set the decoding temperature to zero for all LLMs to obtain reproducible responses. We defer comparison with additional baselines and *LLaMA2* models in Appendix C. and discuss performance on BiTOD dataset in Appendix F. Training details for hint predictors and retrieval are in Appendix D.

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). Across both datasets, *SyncTOD* variants demonstrate competitive Entity F1 scores, with *SyncTOD* (*GPT-4*) outperforming all the supervised baseline models. Further, *ChatGPT* and *GPT-4* enjoy consistent performance gains when coupled with *SyncTOD*.

Interestingly, RAG LLMs display a stronger entity F1 performance on SMD than MultiWOZ. In MultiWOZ and SMD, users express preferences

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
Zero-shot (*ChatGPT*)	3.39	28.16	6.91	60.11
RAG (*ChatGPT*)	8.98	40.2	16.71	70.25
RAG (*GPT-4*)	7.64	41.14	13.44	71.02
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.

Model	MultiWOZ		SMD	
	Relevance	Grammar	Relevance	Grammar
MAKER	4.38	4.79	4.51	4.97
Gold	4.62	**4.9**	4.79	4.95
SyncTOD (GPT-4)	**4.68**	4.8	**4.81**	**4.98**

Table 3: Human evaluation results on MultiWOZ and SMD datasets. Here inter-annotator agreement is Kendall's Tau $\tau = 0.51$ with ($p < 0.001$)

differently. In MultiWOZ, users give detailed preferences for area, price, rating, etc., and can change these during conversation. In SMD, preferences are simpler, like the nearest parking, city weather, or meeting times. Thus, MultiWOZ presents a more challenging problem for LLMs than SMD.

Unlike Entity F1, *SyncTOD* variants do not seem competitive in response quality, as measured by BLEU. Upon analysis, we find that *SyncTOD* responses are meaningful but use alternate phrasing and do not have enough lexical overlap with the gold, thus impacting BLEU scores. We investigate this further in our human evaluation.

Human Evaluation: We task two annotators to evaluate responses from Gold, MAKER[1], and *SyncTOD* (*GPT-4*) models. Specifically, we evaluate model responses for a) *relevance* to the dialog history and KB and b) *grammar* on a 1-5 Likert Scale (Likert, 1932). Appendix H discusses our evaluation protocol in detail. We report our findings in Table 3. *SyncTOD* scores better than MAKER on relevance and grammar across datasets, indicating a superior response quality.

Low Data Setting: Figures 2 and 3 present the evaluation with varying training data sizes. With

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

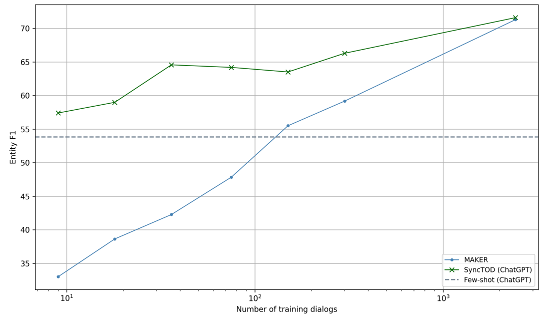


Figure 2: Model performance on SMD dataset at different training data sizes.

	MultiWOZ	SMD
SyncTOD (*ChatGPT*)	52.99	71.60
w\o hint prediction	40.2	70.25
w\o exemplar retrieval	45.47	66.84
w\o exemplar reranking	49.94	71.60

Table 4: Ablation Study: Entity F1 on MultiWOZ and SMD datasets

limited training data, *SyncTOD* (*ChatGPT*) consistently improves *ChatGPT* performance and outperforms MAKER. In MultiWOZ, *SyncTOD* (*ChatGPT*) maintains a lead over the competitors, with MAKER finally catching up at around 1000 dialogs. The gains are even more prominent in SMD, where, with less than 20 examples, *SyncTOD* (*ChatGPT*) achieves Entity F1 comparable to MAKER, trained with 16x more data.

Ablations: We perform ablations on *SyncTOD* (*ChatGPT*) and report results in Table 4. 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 the simpler nature of dialogs in SMD that allows *SyncTOD* to retrieve high-quality exemplars without re-ranking.

5 Conclusion

We propose *SyncTOD* that leverages LLMs for end-to-end TOD. 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. We will release code for future research.

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 be improved by designing much more sophisticated hints.

Ethical Considerations

In this work, we use OpenAI's *ChatGPT* and *GPT-4* which are commercial LLMs whose training details are not publicly available. Thus, it is unclear whether these models have seen the datasets used in this work during their training. In our experiments, we benchmark Zero-shot (*ChatGPT*) on all the datasets and report the performance in table 2. As zero-shot (*ChatGPT*) performs poorly, we believe that our datasets were not part of *ChatGPT*'s training set.

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.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *ArXiv*, abs/2002.08909.

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*.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *ArXiv*, abs/2005.11401.

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.

Rensis Likert. 1932. A technique for the measurement of attitude scales.

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, Peng Xu, Feijun Jiang, Yuxiang Hu, Chen Shi, and Pascale Fung. 2021. Bitod: A bilingual multi-domain dataset for task-oriented dialogue modeling. *arXiv preprint arXiv:2106.02787*.

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.

Nicholas Meade, Spandana Gella, Devamanyu Hazarika, Prakhar Gupta, Di Jin, Siva Reddy, Yang Liu, and Dilek Z. Hakkani-Tür. 2023. Using in-context learning to improve dialogue safety. *ArXiv*, abs/2302.00871.

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.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

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.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *ArXiv*, abs/2301.12652.

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*.

Liang Wang, Nan Yang, and Furu Wei. 2023. Learning to retrieve in-context examples for large language models. *ArXiv*, abs/2307.07164.

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.

Jianguo Zhang, Stephen Roller, Kun Qian, Zhiwei Liu, Rui Meng, Shelby Heinecke, Haiquan Wang, Silvio Savarese, and Caiming Xiong. 2023a. Enhancing performance on seen and unseen dialogue scenarios using retrieval-augmented end-to-end task-oriented system. In *SIGDIAL Conferences*.

Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. 2023b. Retrieve anything to augment large language models. *ArXiv*, abs/2310.07554.

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 using a retrieve-rerank approach.

A natural approach for combining training data with in-context learning is via retrieval-augmented generation (RAG) (Lewis et al., 2020; Guu et al., 2020). Here, a retriever model infuses LLM input with exemplars from the training that are similar to the test sample (Lewis et al., 2020; Meade et al., 2023; Shi et al., 2023; Ram et al., 2023). Although out-of-box retrievers work reasonably well (Ram et al., 2023), many recent works strive to improve the retriever model further. Notably, (Zhang et al., 2023b; Wang et al., 2023) employ reward-based and contrastive learning to improve retrieval quality. Specifically, they use LLMs to obtain soft rewards to fine-tune the retriever model. What sets *SyncTOD* apart from RAG is its use of hints not only for selecting the informative exemplars but also for steering LLM generation from within the prompt.

B Dataset Details

For MultiWOZ and SMD datasets, we use the versions of the dataset released by Wan et al. (2023). We adapt BiTOD dataset (Lin et al., 2021) to end-to-end setting by associating KB to the English dialogs available in the dataset.

Dataset	Domain	#train	#val	#test
MultiWOZ	Restaurant, Hotel, Attraction	1839	117	141
SMD	Navigate, Schedule, Weather	2425	302	304
BiTOD	Restaurant, Hotel, Attraction	1614	169	152

Table 5: Evaluation Dataset Details

C Additional Baselines

We compared our model against the following end-to-end TOD baselines - We compare *SyncTOD* 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 6.

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
Zero-shot (*ChatGPT*)	3.39	28.16	6.91	60.11
Few-shot (*ChatGPT*)	8.83	40.25	17.21	70.58
Few-shot (*GPT-4*)	6.25	36.47	10.08	63.57
RAG (*ChatGPT*)	8.98	40.2	16.71	70.25
RAG (*GPT-4*)	7.64	41.14	13.44	71.02
Few-shot (*LLaMA2* 70B)	5.26	39.68	3.29	46.20
Few-shot (*LLaMA2* Chat 70B)	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 6: 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 MultiWOZ low data setting results

Figure 3 compares the performance of *SyncTOD* (*ChatGPT*), and MAKER on an increasing number of training dialogs from MultiWOZ dataset. Similar to SMD dataset, we find that *SyncTOD* (*ChatGPT*) displays consistent lead over MAKER in low-data setting.

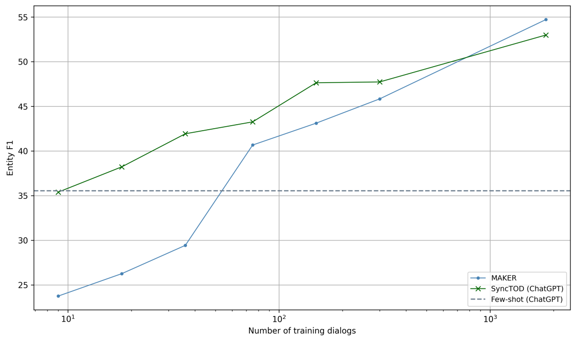


Figure 3: Model performance in low data setting for MultiWOZ dataset.

F Performance on BiTOD dataset

Full-data Setting: Results are shown in table 7. For each reported baseline, we use the code released by the respective authors. As in MultiWOZ and SMD datasets, we observe similar trends for BLEU and Entity F1 where *SyncTOD* (*ChatGPT*) and *SyncTOD* (*GPT-4*) models achieve competitive performance against the SoTA approaches.

Model	**BLEU**	**Entity-F1**
GLMP	23.55	68.87
FG2Seq	32.09	82.91
CDNet	25.49	77.13
DialoKG	27.68	66.98
UnifiedSKG	36.73	88.62
MAKER	32.21	80.00
Zero-shot (ChatGPT)	3.37	38.37
Few-shot (ChatGPT)	12.09	55.50
Few-shot (ChatGPT)	16.67	83.43
RAG (ChatGPT)	10.33	53.62
RAG (GPT-4)	8.09	56.93
SyncTOD (ChatGPT)	19.81	86.04
SyncTOD (GPT-4)	19.34	89.04

Table 7: *SyncTOD* performance on BiTOD dataset.

Low-data Setting: Figure 4 shows compares performance of MAKER and *SyncTOD* (*Chat-*

GPT) at increasing data scales. *SyncTOD* (*ChatGPT*) has significant gains over MAKER across all data scales. With less than 20 examples, *SyncTOD* (*ChatGPT*) achieves Entity F1 comparable to MAKER, trained with 16x more data.

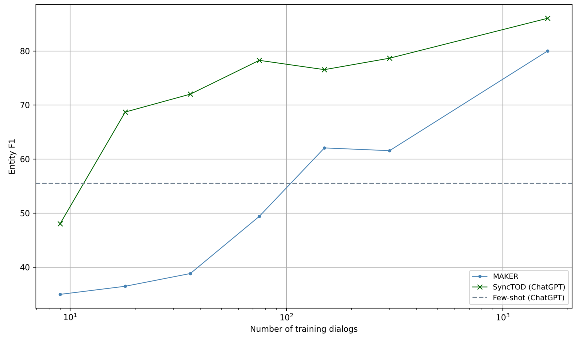


Figure 4: Model performance in low data setting for BiTOD dataset.

Ablation Study on BiTOD dataset is reported in table 8.

	Entity-F1
SyncTOD (*ChatGPT*)	86.03
w\o hint prediction	53.62
w\o exemplar retrieval	63.44
w\o exemplar reranking	78.04

Table 8: Ablation study: Entity-F1 on BiTOD dataset.

G Hint Predictors Performance

Accuracy	MultiWOZ	SMD	BiTOD
Closure Prediction	0.9564	0.9109	0.9570
Entity Type Prediction	0.6805	0.7436	0.8778

Table 9: Accuracy of hint Predictor models.

Table 9 reports the performance of *SyncTOD* hint predictors. We report accuracy for DC predictor and micro F1 for ET predictor. 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.

H Human Evaluation Details

A snapshot of our human evaluation portal is given in figure 5. Detailed evaluation guidelines are given at the end of this section.

In this work, we human-evaluate responses from three TOD systems - Gold, MAKER, and *SyncTOD* (*GPT-4*). We randomly sample 80 dialog context-response pairs from the MultiWOZ dataset. Two annotators, graduate student volunteers, then independently score TOD system responses for these 80 samples on a Likert scale (Likert, 1932) according to evaluation guidelines given below.

Task Overview

There are 80 dialog context response pairs in the html file. Each context response pair dictates a scenario where user is enquiring the agent about hotels, restaurant and attractions to visit.

- User can optionally request for additional attributes like phone number and address and can make a booking.
- Agent is expected to suggest hotel, restaurant and attraction with the highest rating among available options.
- In each scenario, agent re-confirms details like user's name, selected hotel/restaurant/attraction, number of people, rooms and dates before making the final booking.

Along with the context response pair, there are outputs of different dialog systems (randomly shuffled). You are requested to annotate each system generated output along two dimensions: relevance and grammar using the following scale:

1. SA: Strongly Agree
2. A : Agree
3. N : Neutral
4. D : Disagree
5. SD: Strongly Disagree

How to judge relevance?

1. Strongly Agree - when the generated output conveys the intended information –correct entity (hotel/restaurant/attraction) and its attributes (address, phone, rating, etc). Also, when generated output requests correct input from the user.
2. Agree – when generated output contains partial information (e.g., when user request address and phone number but output contains only address).

3. Neutral – when generated output is hard to decide whether its right or wrong.

4. Disagree - when the generated response is somewhat unacceptable (e.g., re-querying already known information like cuisine for restaurants and name of the user for booking).

5. Strongly Disagree – when the generated output contains incorrect information (entities or attributes) for given conversation context.

How to judge grammar?

The grammar of the response is independent of the dialog context or ground truth. A system output can be marked strongly disagree for relevance and still be marked strongly agree for grammar. You can make your own rules about what each rating in the scale means for grammar, but please be consistent with the rules you come up with.

I Case Studies

We present examples from MultiWOZ (table 10), SMD (tables 11 and 12) and BiTOD (table 13) comparing responses from *SyncTOD*, MAKER and the gold systems.

J Prompt Specification

J.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.

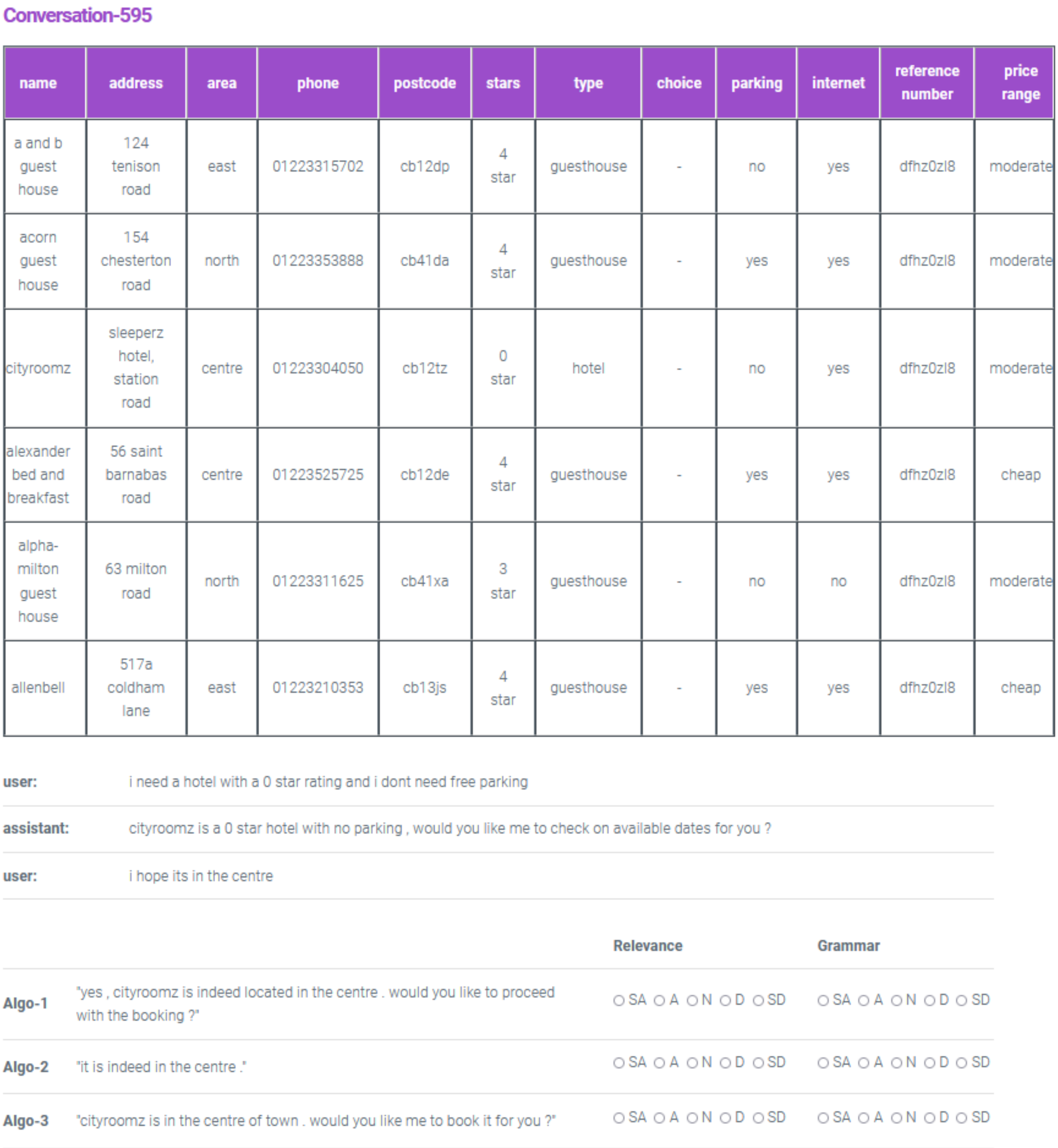
`rules` - We include hints H as a set of rules in the prompt and ask the LLM to follow the rules for writing the response. Rules guide the LLM toward the desired answer. We provide further details on rule creation at the end of this section.

`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.

J.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.


Conversation-595

name	address	area	phone	postcode	stars	type	choice	parking	internet	reference number	price range
a and b guest house	124 tenison road	east	01223315702	cb12dp	4 star	guesthouse	-	no	yes	dfhz0zl8	moderate
acorn guest house	154 chesterton road	north	01223353888	cb41da	4 star	guesthouse	-	yes	yes	dfhz0zl8	moderate
cityroomz	sleeperz hotel, station road	centre	01223304050	cb12tz	0 star	hotel	-	no	yes	dfhz0zl8	moderate
alexander bed and breakfast	56 saint barnabas road	centre	01223525725	cb12de	4 star	guesthouse	-	yes	yes	dfhz0zl8	cheap
alpha-milton guest house	63 milton road	north	01223311625	cb41xa	3 star	guesthouse	-	no	no	dfhz0zl8	moderate
allenbell	517a coldham lane	east	01223210353	cb13js	4 star	guesthouse	-	yes	yes	dfhz0zl8	cheap

user: i need a hotel with a 0 star rating and i dont need free parking

assistant: cityroomz is a 0 star hotel with no parking , would you like me to check on available dates for you ?

user: i hope its in the centre

		Relevance	Grammar
Algo-1	"yes , cityroomz is indeed located in the centre . would you like to proceed with the booking ?"	○ SA ○ A ○ N ○ D ○ SD	○ SA ○ A ○ N ○ D ○ SD
Algo-2	"it is indeed in the centre ."	○ SA ○ A ○ N ○ D ○ SD	○ SA ○ A ○ N ○ D ○ SD
Algo-3	"cityroomz is in the centre of town . would you like me to book it for you ?"	○ SA ○ A ○ N ○ D ○ SD	○ SA ○ A ○ N ○ D ○ SD

Figure 5: Human Evaluation Portal

Name	Address	Area	Phone	Postcode	Pricerange	Stars	Type	Choice	Ref	Parking	Internet
alpha-milton guest house	63 milton road	north	01223311625	cb41xa	moderate	3 star	guesthouse	-	epwuawm5	no	no
alexander bed and breakfast	56 saint barnabas road	centre	01223525725	cb12de	cheap	4 star	guesthouse	-	epwuawm5	yes	yes
acorn guest house	154 chesterton road	north	01223353888	cb41da	moderate	4 star	guesthouse	-	epwuawm5	yes	yes
allenbell	517a coldham lane	east	01223210353	cb13js	cheap	4 star	guesthouse	-	epwuawm5	yes	yes
bridge guest house	151 hills road	south	01223247942	cb28rj	moderate	3 star	guesthouse	-	epwuawm5	yes	yes
a and b guest house	124 tenison road	east	01223315702	cb12dp	moderate	4 star	guesthouse	-	epwuawm5	no	yes

Context	**User:** I' m looking for a place. I think it's called like bridge house or bridge guesthouse or something.
Gold	You are looking for the Bridge Guest House.
MAKER	I'm sorry, I don't have any information on that. Would you like to try a different name?
SyncTOD (*ChatGPT*)	Bridge guest house is a moderate priced guesthouse in the south with a 3 star rating. Can I help you book a room there?
SyncTOD (*GPT-4*)	Yes, the Bridge Guest House is a 3 star guesthouse in the south with moderate pricing.

Table 10: *SyncTOD* models understand user's requirement for *bridge guest house* and present required information from the KB. MAKER, however, produces incorrect results.

Event	**Room**	**Agenda**	**Time**	**Date**	**Party**
conference	conference room 102	go over budget	7pm	monday	boss
lab appointment	-	-	9am	tuesday	tom
dinner	-	-	7pm	tuesday	martha
optometrist appointment	-	-	5pm	the 5th	brother

Context	**User:** I need to know my conference agenda, date and who is attending. **System:** Your conference is set for Monday with your boss to go over budget. **User:** Great, and who all will be there?
Gold	The conference is with your boss.
MAKER	You have three conference agendas, one for Monday with your boss to go over budget, one for Friday with your boss to go over the budget, and one for Monday with your boss to go over the budget.
SyncTOD (*ChatGPT*)	Your boss will be attending the conference.
SyncTOD (*GPT-4*)	Your boss will attend the conference.

Table 11: *SyncTOD* responses are grounded into KB while MAKER provides repeated and incorrect information.

Poi	Distance	Address	Poi Type	Traffic Info
Coupa	6 miles	394 van ness ave	coffee or tea place	moderate traffic
Mandarin Roots	2 miles	271 springer street	chinese restaurant	heavy traffic
Four Seasons	3 miles	465 arcadia pl	rest stop	no traffic
Ravenswood Shopping Center	1 miles	434 arastradero rd	shopping center	moderate traffic

Context	**User:** Give me directions to the local starbuck **System:** I do not have info on the local starbucks but i do see another coffee place. Would you like to hear info on it? **User:** Yes, please! Which one is its address? There is any quick route available?
Gold	There is Coupa located at 394 van ness ave, there is moderate traffic but you're just 6 miles away
MAKER	I'm sorry, I do not have that info on you, I sent it on your screen, drive carefully!
SyncTOD (*ChatGPT*)	Coupa is located at 394 van ness ave. There is moderate traffic on the way.
SyncTOD (*GPT-4*)	Coupa at 394 van ness ave has moderate traffic.

Table 12: *SyncTOD* models accurately answer user's query.

	name	address	cuisine	dietary_restrictions	location	phone_number	price_level	rating
0	chocoduck_bistro	3_kin_san_lane_cheung_chau	italian	vegan_options	new_territories	852_9735_0049	moderate	10
1	la_table_french_brasserie	72_mody_road_1f_new_world_millennium_hong_kong...	french	vegan_options	tsim_sha_tsui	852_2313_4222	moderate	10
2	fleur_de_sel	shop_2j_po_foo_building_foo_ming_street_causew...	french	vegan_options	wan_chai_causeway_bay	852_2805_6678	moderate	10
3	serendib	shop_2_gf_nam_wo_hong_building_148_wing_lok_st...	asian	vegan_options	sheung_wan	852_3705_2429	moderate	10
4	bombay_dreams	77_wyndham_street	indian	vegan_options	central	852_2971_0001	moderate	9
5	divino_wine_bar_restaurant	73_wyndham_street_central	italian	vegan_options	central	852_2167_8883	moderate	9
6	enchanted_garden_restaurant	hong_kong_disneyland_resort_disneyland_hotel_l...	seafood	vegan_options	new_territories	852_3510_6000	moderate	9
7	jw_cafe	5f_jw_marriott_hotel_hong_kong_pacific_place_n...	cafe	vegan_options	hong_kong_island	852_2810_8366	moderate	9

Context	**User:** Hello. Can you help me find a place to eat ? **System:** Would you like to choose the rating value for the restaurant? **User:** Not really . I just want to make sure they have vegan_options . **System:** Did you have a price range or cuisine type in mind ? **User:** Any cuisine will work , but it needs to be moderately priced . **System:** I've found #210 options with those criteria . May i recommend an Italian place with a rating of 10 called chocoduck_bistro ? **User:** That sounds good . can i make a reservation ? **System:** Definitely ! how many will be in your party and what time would you like to book ? **User:** There will be 14 of us , and we can be there by 410 in the afternoon on Sunday . **System:** Wonderful , and what name should i put that under ? **User:** Put it under Danielle , please .
Gold	Great ! Just to confirm your reservation for Sunday at 410 in the afternoon at chocoduck_bistro for a party of 14 , correct ?
MAKER	Okay , and what is your booking time ?
SyncTOD (*ChatGPT*)	Got it, I can make a reservation for 14 people at chocoduck_bistro on Sunday at 410 pm under the name Danielle.
SyncTOD (*GPT-4*)	So, a reservation for 14 at chocoduck_bistro, 410 pm on Sunday.

Table 13: *SyncTOD* models assists user in making the reservation.

J.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 -

BiTOD

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 number - phone number of the place
4. location - a part of the city e.g. canal road, central district
5. rating - user rating of the place out of 10 e.g. 8, 9
6. price level - price range of the place, e.g. cheap, moderate, expensive
7. reference number - reference code for booking, e.g. 542j9wog
8. stars - star rating of the hotel, e.g. 3 stars
9. price per night - hotel charges per night e.g. 512, 600, etc.
10. number of rooms - number of rooms to book for the customer e.g. 1, 2
11. number of nights - number of nights the customer wants to book the hotel e.g. 2, 3
12. user name - name of the user e.g. Jack, Henry
13. start month - starting month of the booking e.g. July, May, etc.
14. start day - starting day of the booking e.g. 12, 30 etc.
15. cuisine - the cuisine of a restaurant, e.g. thai, chinese, etc.
16. dietary restrictions - dietary restrictions that the restaurant facilitates e.g. vegan, gluten free
17. number of people - number of people to reserve a restaurant for e.g. 2, 10
18. month - a month of the year e.g. january, february, etc.
19. day - a day of the month/week e.g. 12, 17, monday, etc.
20. time - a time of the daay e.g. 1200, 1330, 930, etc.
21. type - type of an attraction e.g. zoos and aquariums, shopping, 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]

```
"house 1881":
  "phone number":"852 0071 5353",
  "location":"tsim sha tsui",
  "rating":"8",
  "price level":"expensive",
  "reference number":"swm2n2uu",
  "stars":"5",
  "price per night":"1895",
  "number of rooms":"2"
,...
```

[rules 1]
The response must be 20 words or shorter.
The response must not close the dialog.
The response must only include entities of type - name, rating.
The response must not include any entities of type - address, phone number, location, price level, reference number, stars, price per night, number of rooms, number of nights, user name, start month, start day, cuisine, dietary restrictions, number of people, month, day, time, type.

[dialog history 1]
user: hi , i'm looking for recommendations for hotels .
assistant: sure , what hotel rating are you looking for ?
user: i want at least a rating of 1 .
assistant: ok , what about price level or location ?
user: i want a expensive hotel . i don't care about the location .
assistant: sounds good , what about the number of stars of the hotel ?
user: should be at least 2 stars .

[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 ['name', 'rating'] in my response.
I will include these entities - [['name', 'house 1881'], ['rating', '8']]
assistant: i found #16 hotels that are available . the house 1881 looks good . it has a rating of 8 .

[example 2]
[database 2]

```
"house 1881":
  "phone number":"852 0071 5353",
  "location":"tsim sha tsui",
  "rating":"8",
  "price level":"expensive",
  "reference number":"swm2n2uu",
  "stars":"5",
  "price per night":"1895",
  "number of rooms":"2"
,...
```

[rules 2]
The response must be 19 words or shorter.
The response must not close the dialog.
The response must only include entities of type - name, rating.
The response must not include any entities of type - address, phone number, location, price level, reference number, stars, price per night, number of rooms, number of nights, user name, start month, start day, cuisine, dietary restrictions, number of people, month, day, time, type.

[dialog history 2]
user: hey ! i am looking for hotels with at least 2 stars . do you have any recommendations ?
assistant: glad to be of service . to get started , can you tell me what rating level and price range are you looking for ?
user: oh yeah ! i am looking for an expensive hotel with minimum 4 rating .

[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 ['name', 'rating'] in my response.
I will include these entities - [['name', 'house 1881'], ['rating', '8']]
assistant: as per your needs , there are #16 hotels available . i would recommend house 1881 with rating 8 .

[example 3]
[database 3]

```
"jw marriott hotel hong kong":
  "phone number":"852 7885 6633",
  "location":"hong kong island",
  "rating":"9",
  "price level":"expensive",
  "reference number":"s5y9h2s3",
  "stars":"5",
  "price per night":"2210",
  "number of rooms":"10"
,...
```

[rules 3]
The response must be 13 words or shorter.
The response must not close the dialog.
The response must only include entities of type - name, rating.
The response must not include any entities of type - address, phone number, location, price level, reference number, stars, price per night, number of rooms, number of nights, user name, start month, start day, cuisine, dietary restrictions, number of people, month, day, time, type.

[dialog history 3]
user: hello . i'm trying to find a hotel for my stay with at least 4 stars . would you be able to help me ?
assistant: hi there . i would be happy to help . would you like an expensive or affordable priced hotel ? do you prefer a high rating hotel ?
user: i would like an expensive hotel with a rating of at least 4 .

[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 ['name', 'rating'] in my response.
I will include these entities -