

Enhancing Task-Oriented Dialog System with Subjective Knowledge: A Large Language Model-based Data Augmentation Framework

Haein Jung* Heuiyeen Yeen* Jeehyun Lee* Minju Kim* Namoo Bang Myoung-Wan Koo

Department of Artificial Intelligence, Sogang University, Korea

{haeindain, yeen214, jhlee22, 0307mjk, namoo950815, mwkoo}@sogang.ac.kr

Abstract

As Task-Oriented Dialog (TOD) systems have advanced, structured DB systems, which aim to collect relevant knowledge for answering user's questions, have also progressed. Despite these advancements, these methods face challenges when dealing with subjective questions from users. To overcome this, DSTC11 released a subjective-knowledge-based TOD (SK-TOD) dataset and benchmark. This paper introduces a framework that effectively solves SK-TOD tasks by leveraging a Large Language Model (LLM). We demonstrate the proficient use of LLM for each sub-task, including an adapters-based method and knowledge-grounded data augmentation. Our proposed methods, which utilize LLM as an efficient tool, outperform baseline performance and approaches that directly use LLM as a one-step sub-task solver, showing superior task-specific optimization.

1 Introduction

In many Task-Oriented Dialog (TOD) systems, to ensure accurate responses to user inquiries, it is often necessary to generate system responses by extracting relevant information from a preprocessed database (Liu and Lane, 2017; Zhong et al., 2018). However, this method shows a generalization limitation, making it difficult to extend to other domains or new information. To overcome this limitation, research has been conducted to generate system responses based on factual information using Frequently Asked Questions (FAQs), a readily available source of factual information (Kim et al., 2022).

However, DSTC11 Track5¹ presents a new limitation that previous research and datasets cannot handle users' subjective requests (e.g. "Do they have nice outdoor dining area?"). Therefore, they have released a subjective-knowledge-based TOD

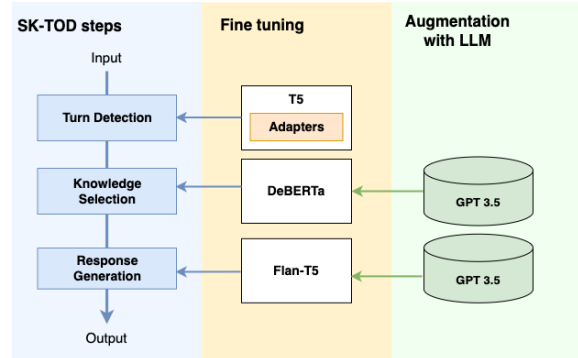


Figure 1: Diagram for each step of solving the SK-TOD task. The leftmost (blue) section represents the overall pipeline for addressing SK-TOD. The middle (orange) section shows the types of models fine-tuned specifically for each task. The rightmost (green) section indicates the augmentation process enhanced through LLM to improve the performance of the fine-tuned models.

(SK-TOD) dialog dataset, which allows for appropriate subjective knowledge (reviews) for user's subjective requests (Zhao et al., 2023).

DSTC11 Track5 defines three sub-tasks to generate appropriate system responses through the SK-TOD dataset. (1) Finding knowledge-seeking turns, (2) Selecting knowledge (reviews or FAQs) based on the dialog history of each knowledge-seeking turn, and (3) Generating system responses based on the selected knowledge and dialog history. The baseline of DSTC11 Track5 selects optimized models for each pipeline and fine-tunes them on the given dataset. However, about 50 percent of the total dataset requires knowledge selection turns, so from Task2 onwards, the dataset for the learning process decreases, which may similarly result in performance degradation.

In particular, recent research shows that Large Language Models (LLMs) demonstrate excellent generalization performance across various tasks (OpenAI, 2023), including conversation-related tasks (Zhang et al., 2023), even in few-shot or zero-shot scenarios. However, a disadvantage of LLMs

*Equal contribution, co-first authors

¹<https://github.com/alexadstc11-track5>

is that optimizing for specific tasks or domains is difficult.

Therefore, this paper proposes a framework that primarily fine-tunes task models suitable for each sub-task, while efficiently utilizing LLMs for performance improvement. By researching the use of LLMs ideal for the characteristics of each stage, it is shown that the mentioned limitations can be overcome by using LLMs and fine-tuning models appropriately. As a result, it has outperformed the benchmark baseline performance in all three sub-tasks of the officially released DSTC11 Track5 test set.

To summarize the contributions of this paper:

- It presents how to use LLMs suitable for the sub-tasks of the SK-TOD task: knowledge-seeking turn detection, knowledge selection, and response generation, and proposes a framework for learning by appropriately integrating LLMs and task models.
- It releases datasets augmented with LLM for each task. These datasets can be utilized for further research.

2 Task Description

The detailed formulation of the SK-TOD Task in DSTC11 Track5 follows the paper (Zhao et al., 2023). A formal dialog context $C = [U_1, S_1, U_2, S_2, \dots, U_t]$ is given between the user and the system. Each user utterance U_i is followed by the system response utterance S_i , except the last user utterance U_t . The dialog is accompanied by subjective background knowledge $B = [(e_1, R_1), (e_2, R_2), \dots]$, which consists of one or more entities $E = [e_1, \dots, e_m]$ and their corresponding customer reviews R . Each entity e can have multiple reviews $R = [R_1, R_2, \dots]$, which can be divided into segments $[K_1, K_2, \dots]$ such as paragraphs, sentences, or sub-sentences.

Therefore, each sub-task can be redefined as follows:

1. **Knowledge-Seeking Turn Detection (KTD):** Determine if the last user utterance U_t in the given dialog requires knowledge access.
2. **Knowledge Selection (KS):** For the turns that require knowledge access, extract the entities (hotel names, restaurant names, etc.) appearing in those turns using a word-matching-based approach. Based on the extracted enti-

	Train	Vaild	Test
Full Data	28,431	4,173	5,475
Knowledge-Seeking Data	14,768	2,129	2,798

Table 1: Dialog instance statistics for the SK-TOD dataset.

ties and the dialog history C , extract relevant knowledge snippets K^+ .

3. **Response Generation (RG):** Generate the final system response S_t based on the extracted knowledge.

The statistics of the publicly available dataset for training, validation, and testing are shown in Table 1.

3 Methodology

3.1 Knowledge-Seeking Turn Detection

To determine if knowledge is required, we apply a sequence generation approach with a language modeling objective. The model predicts the next token in the dialog context and is trained with categorical cross-entropy loss to generate either "True" or "False" as the output

Adapter Tuning We leverage a pre-trained language model, which was trained on a large-scale dialog dataset, using an adapter-based approach. Adapter modules (Houlsby et al., 2019) are integrated into each transformer layer, keeping the base model’s parameters frozen. These adapters consist of two feed-forward layers that project the feature size into a reduced dimension and then restore it to the original dimension. Throughout the training process, we freeze the pre-trained layers and exclusively train the adapters. One key advantage of using adapters is their ability to mitigate the catastrophic forgetting problem, enabling the model to preserve knowledge from previously learned tasks while adapting to new ones. We focus on retraining the encoder and decoder layer adapters specifically for the KTD task while preserving pre-trained knowledge.

3.2 Knowledge Selection

3.2.1 Entity Tracking

To streamline the pool of potential knowledge candidates, we implement entity matching as delineated in Zhao et al., 2023. The goal is to identify an entity that is relevant to both the dialog history

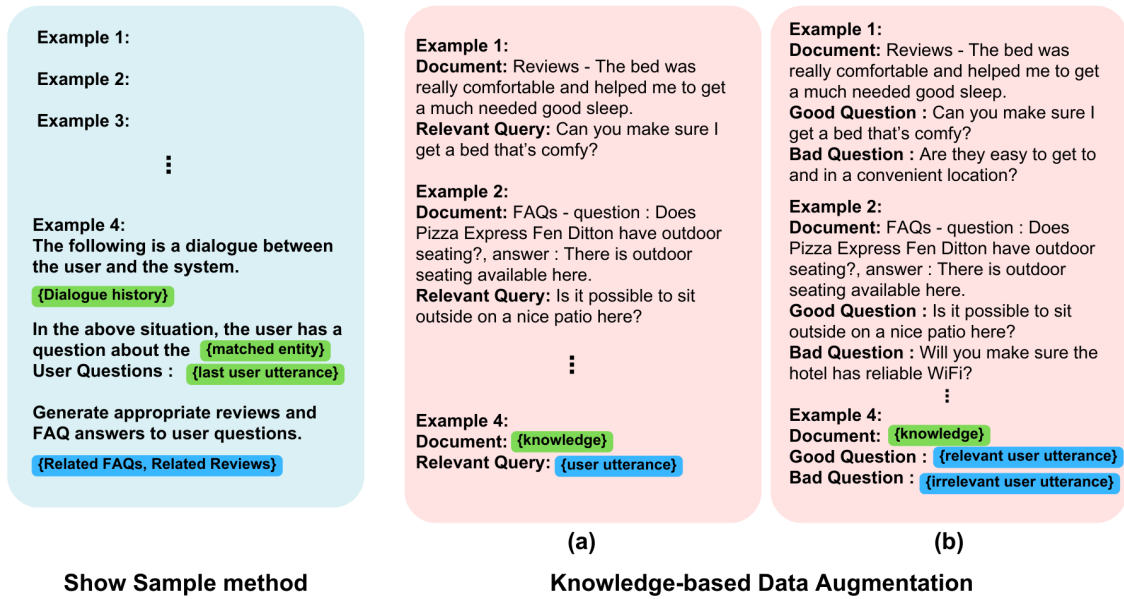


Figure 2: Example prompts for the *Show Sample Method* and *Knowledge-based Data Augmentation* methods used in the KS task. The bold text represents the default prompt format, and the input provided is highlighted in green. The blue text denotes the portion generated by the model.

and the user request among the potential candidates. Therefore, the matching process is carried out based on a word-matching-based methodology.

3.2.2 Data Augmentation for Knowledge Selection

During the training process, positive (C, K) pairs are constructed using all relevant knowledge snippets, and an equal number of negative pairs are randomly sampled. The model is trained to minimize the binary cross-entropy loss. Therefore, it is necessary to construct various positive pairs depending on the case to find appropriate K^+ for a given C . We leverage LLM to improve performance by transforming and augmenting datasets.

Show Sample Method In Wang et al., 2022, generating rationales for given questions and fine-tuning them through question matching with a small model enhanced performance. This approach outperforms alternatives like using prompts in existing QA tasks or direct fine-tuning. Incorporating pre-training knowledge from the LLM enables more enriched training.

We follow the approach (Wang et al., 2022) to generate FAQs and reviews, instead of rationales, which can represent the dialog history, last user utterance, and the matched entity. The prompt includes 2-shot exemplars so that it could be created

in a similar way. The actual prompt is structured as shown in Figure 2. It is concatenated with the last user utterance and fed into the model for training. In actual inference, the dialog history is first input to the LLM, and representative samples are generated. The inference is then conducted by appending these generated samples to the last user utterance.

Knowledge-based Data Augmentation From a different perspective, our work draws inspiration from the paper (Bonifacio et al., 2022), which introduces a data augmentation method for training a ranker that extracts relevant knowledge. In this context, we adapt a similar approach to generate user’s subject requests matching each knowledge (reviews, FAQs) attached to all entity lists.

Two prompting methods are employed in the experiments, as depicted in Figure 2. In method (a), relevant queries are generated based on the provided knowledge (K) . On the other hand, method (b) generates both relevant and irrelevant queries. In both prompting methods, 3-shot exemplars are given. For method (b), only the portion corresponding to a relevant query is extracted and used after generation. The generated utterances are labeled as single-turn dialogs requiring knowledge access and are associated with the underlying knowledge as the corresponding labels.

	BLEU	MT	R-1	R-2	R-L
Flan-T5-XL	9.61	17.15	35.72	14.67	27.98
+ pre-train (SM)	10.19	17.42	35.91	14.55	27.85
+ pre-train (LR)	10.16	18.68	37.16	15.22	28.72

Table 2: Comparison of pre-train methods on the RG task. SM stands for span masking, and LR stands for last response. Also, MT stands for METEOR metric, and R stands for Rouge metric.

3.2.3 Fine-tuning

For the cross-encoder approach, we encode the concatenation of C and K to obtain the contextualized representation. The actual training is conducted through binary cross-entropy loss by combining positive and negative pairs based on labels, identical to the paper (Zhao et al., 2023).

$$h = Enc(C, K) \quad (1)$$

$$P(C, K) = softmax(FFN(h)).$$

3.3 Response Generation

RG tries to answer the user’s question accurately based on the reviews or FAQs, after the knowledge snippets ($K+$) retrieved from the KS stage and the dialog context. In this paper, we explore two methodologies to study how LLM can be effectively used to generate responses: a method of fine-tuning the Flan-T5-XL model (Wei et al., 2021), and in-context learning (Brown et al., 2020) using GPT-3.5-turbo model from Open AI².

3.3.1 Fine-tuning

Various Instruction Templates for Fine-tuning

We use Flan-T5 model for fine-tuning because it was developed to understand and respond to natural language instructions, and it helps understand the context of the user’s conversation and conduct knowledge-based QA. Since this model is vital in understanding instructions better, we expect to get better fine-tuning results by using specific instruction templates. We experiment with various instruction templates, and instruction of fine-tuning with span masking tokens works best. The specific prompts we experiment with can be found in appendix B.

Dialog Domain Adaptive Pre-train Since Flan-T5 is a model optimized for instruction rather than conversation, we perform further domain adaptive pre-train with dialog datasets to better understand

the context of the conversation and generate answers naturally (Zhang et al., 2019; Mehri et al., 2019). We use the pre-training corpora which was used to train PPTOD (Su et al., 2021, 2022), and we perform pre-train in two ways: learning with span masking and generating only last response to the last question. For span masking, 15% of the tokens are randomly selected and masked. As described in the table 2, we find that last response generation pre-train task performs better. It seems the pre-train and fine-tuning tasks are similar in that they generate the final response based on the conversation.

Data Augmentation Methods In order to train the model on a larger amount of data, we adopt data augmentation techniques proposed by Dai et al., 2023; Wang et al., 2023. Our focus is on effectively augmenting data using LLM and achieving higher performance compared to using LLM directly as a problem solver. We propose three data augmentation approaches. The first is to augment the data by masking and changing some words, the second is to paraphrase the entire conversation, and the third is to augment the conversation based on knowledge. Appendix C gives details.

First, we adopt the inpaint method proposed in the paper like Wei and Zou, 2019. It is important to preserve the speech style and format of the dataset while augmenting dialogs, knowledge, and system responses through filling in the blanks. As part of the inpaint method, we experiment with two data augmentation techniques: synonym replacement and masked language modeling approach. Inspired by the synonym replacement method introduced in Wei and Zou, 2019, we randomly select one adjective and adverb from each utterance and replace them with synonyms. We use WordNet³’s synonym list. Similarly, we explore the masked language modeling approach (Gao et al., 2023) by masking candidate indices from each utterance, making predictions, and augmenting the data based on these predictions. We utilize a pre-trained XLM-Roberta-Large model (Conneau et al., 2020). By doing so, we introduce diversity and contextually relevant variations in the dataset while maintaining the underlying speech style and format.

Second, we employ the back-translation (Sennrich et al., 2016) method, which changes an existing utterance from the source language to a target language and then translates it back to the original

²<https://platform.openai.com/docs/models/gpt-3-5>

³<https://wordnet.princeton.edu/>

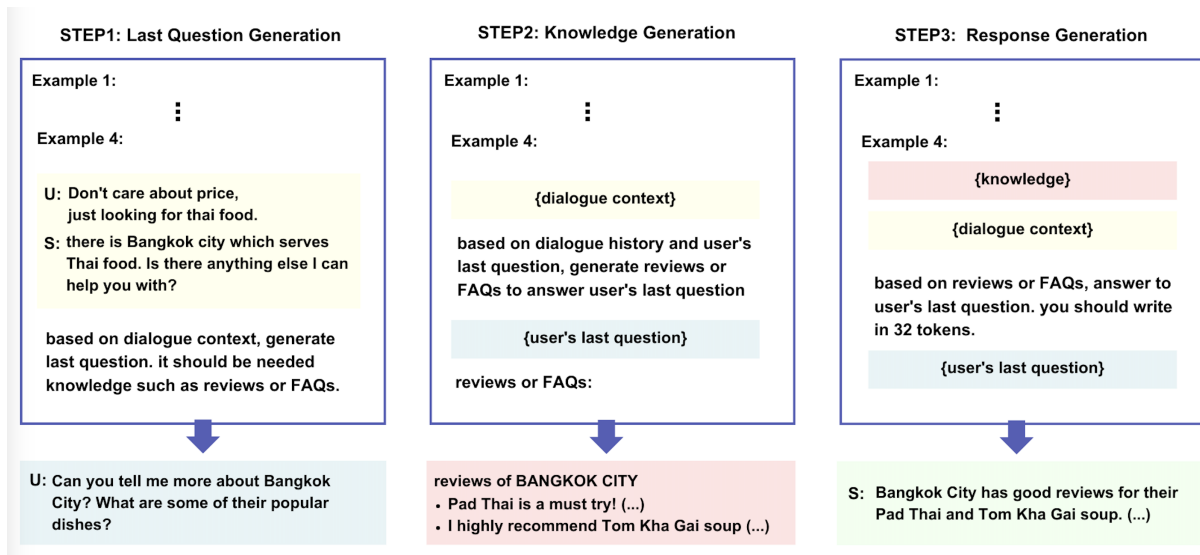


Figure 3: Knowledge-grounded data augmentation methods using GPT-3.5-turbo model.

language. We utilize the `nlpaug`⁴ package for inference, which uses `fairseq`⁵'s pre-trained model to perform the translation. We also experiment with a paraphrase method using pre-trained language model (Gao et al., 2020; Niu et al., 2021; Chowdhury et al., 2022) to generate utterances that have the same meaning and context as existing sentences while ensuring fluency and grammatical correctness. We use the Parrot package (Damodaran, 2021), which provides a paraphrase-based utterance augmentation framework specialized for natural language understanding tasks, and the PEGASUS (Zhang et al., 2020) model, a self-supervised encoder-decoder-based pre-training transformer structure for large documents.

Third, we perform knowledge-grounded augmentation using the GPT-3.5-turbo model as described in figure 3. First, we input the dialog context except the user's last question to GPT, and then use knowledge to generate the user's question that needs to be answered. Next, we input the dialog context and the newly augmented user question, and generate the reviews or FAQs needed to answer it. Finally, we add the augmented knowledge, the dialog context, and the user's last question and have the system to generate a response using based on knowledge. All three stages of data augmentation are performed using 4-shot exemplars.

⁴<https://github.com/makcedward/nlpaug>

⁵<https://github.com/facebookresearch/fairseq>

3.3.2 In-context Learning

Fine-tuning methodologies achieve high performance, but they also require a lot of data and training resources. (Brown et al., 2020) Therefore, we conduct experiments for in-context learning using GPT-3.5-turbo model and try additional prompt engineering methods.

First, we conduct experiments to find the optimal prompt using the manual prompt method. Through this, we find that bundling knowledge or dialog context in the form of python's dictionary data structure generated better responses even if it takes several few-shot exemplars. It seems like GPT understands data better in code structure. In particular, it is the most effective way to present specific instructions at the end, so we put the last question and instruction at the end as described in appendix D.

Second, we use clue-based generation prompts, which is an idea to overcome GPT's strict answers and its inability to ask probing questions based on user intent. GPT generates clues from the dialog context, such as (1) what the user wants, (2) what information should be provided to the user, and (3) what can be asked of the user. GPT then generates an answer by adding the clue, knowledge, and the user's last question.

Third, we use the summary-clue generation method. Similar to the second method, GPT creates a summary-clue that describes the situation, and reinserts this summary-clue when generating a response.

	Precision	Recall	F1
BERT	99.75	99.61	99.68
RoBERTa	99.86	99.64	99.75
ALBERT	99.64	99.36	99.50
DeBERTa	99.86	99.57	99.71
Baseline	99.82	99.79	99.80
Ours	99.54	99.93	99.73

Table 3: Results of the KTD task.

4 Experiments

4.1 Knowledge-Seeking Turn Detection

4.1.1 Experimental Setup

Metrics We report the Precision, Recall, and F1-score for the KTD task.

Model & Hyper-parameters As a starting point, we use the base version of the PPTOD (Su et al., 2022) checkpoint, which has been pre-trained on a large-scale multi-turn task-oriented dialog dataset. For optimization, We use Adafactor (Shazeer and Stern, 2018) with a learning rate of 0.00001. Throughout the training process, we maintain a batch size of 32. We conduct 19 epochs of training exclusively for the adapter modules. The bottleneck dimension of the adapters is set to half of the hidden dimension of the T5 model following Bang et al., 2023. We adopt the baseline provided by the DSTC11 organizer, and the remaining results are from the task introductory paper (Zhao et al., 2023).

4.1.2 Results

The results of the KTD task on the test set are presented in Table 3. Our proposed model, denoted as "Ours," achieves competitive performance with a F1-score of 99.73%. It outperforms all other models with a Recall of 99.93%. These results highlight the model’s ability to detect and classify actual knowledge seeking turns in the dialog context. By leveraging the advantages of adapters, we could enhance the model’s performance, effectively addressing the challenges of the KTD task.

4.2 Knowledge Selection

4.2.1 Experimental Setup

Metrics Precision, Recall, and F1-score have been computed at the knowledge snippet level. Instead of calculating the P/R/F1 for each dialog at the snippet level, the P/R/F1 is calculated for all (C, K) pairs in the entire dataset. Additionally, the exact matching score is used as an additional met-

Version	# of Data	Precision	Recall	F1	Exact Match
Baseline	28,431	79.01	78.77	78.89	39.06
Ver1	28,431	54.78	93	68.95	28.19
Ver2	39,313	77.66	83.12	80.3	42.54
Ver3	39,313	75.33	86.53	80.54	43.36
Ver4	50,195	83.35	82.17	82.76	47.71
Ver5	71,959	78.34	86.84	82.37	47.37

Table 4: Results of the KS task by training data version.

ric, which measures the ratio of dialogs which all knowledge snippets are exactly extracted from the total number of dialogs.

Model & Hyper-parameters The LLM used for creating and augmenting the dataset is the GPT-3.5-turbo model released by Open AI. The model used for fine-tuning the selection task is the DeBERTa (He et al., 2020). For fine-tuning, we set the AdamW optimizer (Kingma and Ba, 2014) as the optimizer, with a train batch size of 4, learning rate of $3e-5$, and max sequence length of 512. The GPU used for training is a single NVIDIA RTX A5000 24G.

Dataset The datasets and versions used in the experiment are in total five, as shown in the Table 4. **1)** The version where FAQs and reviews representative of the dialog are attached to the last user utterance using the show sample method. **2)** The original dataset + knowledge-based data augmentation prompt style (a). **3)** The original dataset + knowledge-based data augmentation prompt style (b). **4)** The original dataset + version 2 + version 3. **5)** version 4 + The data where the history of single user utterances generated in versions 1 and 2 was created and appended.

4.2.2 Results

We compare the performance of the models trained on the dataset built through the LLM with the baseline released in DSTC11 Track5. In fact, version 1 achieves first place in the *Recall* metric part of the DSTC11 Track5 benchmark, and other versions mostly shows higher performance than the baseline. In particular, All scores for versions 2-5 increase significantly compared to the baseline. The overall evaluation results are shown in Table 4. The followings are the points of analysis for these results.

4.2.3 Analysis

Generating Representative Samples & Diverse Examples The first method involves creating representative reviews and FAQs for each given dialog instance, allowing us to build a dataset that can retrieve similar types of knowledge. While this

approach results in many similarly structured or typed reviews and FAQs, we believe its accuracy score is lower due to the generated samples failing to encompass all the relevancy of the actual labels pertaining to knowledge.

On the other hand, referring to the dataset gathering process in the paper (Zhao et al., 2023), they initially created FAQs and user persona-based reviews for 33 hotels and 110 restaurants, then constructed a new database. Subsequently, they inserted a subjective user request that aligned with the existing conversation. This method is very similar to the knowledge-based data augmentation presented in our paper. Therefore, we believe our generation method contributes significantly to performance improvement by showing the model examples with a similar distribution to the original dataset and providing diverse cases.

Presence or Absence of History in Generated Examples For dataset version 2-4, as a single last user utterance is created that corresponds to one piece of knowledge, the dataset has been augmented with single utterance-style data that lacks history. Accordingly, version 5 is also devised to generate the history preceding these newly created utterances. The generated history is limited to start with the user and end with the system, with no more than five utterances. Detailed prompting examples are provided in Appendix A.

Looking at Table 4, adding history in this way to the dataset used in version 4 results in lower precision performance, despite having the most data. However, it does show a higher Recall score. In fact, in the RG stage following the KS, having a higher Recall performs much better when the F1 score is the same (as having more knowledge candidates leads to better generation performance), so ultimately, we use the model trained with version 5 in the RG stage.

4.3 Response Generation

4.3.1 Experimental Setup

Metric For automatic evaluation, we use BLEU, METEOR, ROUGE-1, ROUGE-2, and ROUGE-L to measure the model-generated response alignment with the gold response. In addition, human evaluation is conducted to compensate for the fact that automatic evaluation is not always reliable. (Zhao et al., 2023)

Model & Hyper-parameters Flan-T5-XL model is the backbone for fine-tuning. The dialog domain

Fine-tuning and Data Augmentation Methods					
	BLEU	MT	R-1	R-2	R-L
Baseline	10.04	17.48	35.20	14.30	27.53
Flan-T5-XL	9.61	17.15	35.72	14.67	27.98
Synonym Replacement	10.22	17.18	35.63	14.28	27.93
Filling Mask	10.01	17.28	35.77	14.33	27.83
Back-translation	10.15	17.20	35.85	14.49	28.00
Paraphrase (Parrot)	10.15	17.39	35.57	13.98	27.52
Paraphrase (PEGASUS)	10.09	17.10	35.59	14.29	27.77
ChatGPT Augmentation	10.35	19.01	37.91	15.42	29.37
In-context Learning Results					
	BLEU	MT	R-1	R-2	R-L
Manual Prompt	6.26	17.74	33.70	11.81	24.47
Clue-based Generation	6.36	17.15	33.11	11.38	24.11
Summary-Clue Generation	6.78	17.18	34.14	12.02	24.87

Table 5: Results of the RG task.

further pre-train is performed on 8 NVIDIA A100 80G, with a max length of 256, batch size of 4, learning rate of 1e-4, and AdamW optimizer. For fine-tuning with publicly available train sets and augmented data, we use the Adam optimizer with a max length of 512, a batch size of 4, and a learning rate of 1e-4. We utilize the GPT-3.5-turbo model for additional data augmentation. The baseline follows DSTC11 Track5 method.

4.3.2 Results

The results for the RG task are summarized in Table 5. The table is organized into the fine-tuned part at the top, and the in-context learning part without fine-tuning at the bottom. It can be seen that the performance of the fine-tuned part is higher than the in-context learning part, which can be attributed to the better learning of the benchmark’s speech style and format.

To enhance performance of fine-tuning models, we conduct dialog domain-adaptive pre-train and data augmentation. The rows below the general fine-tuning are the results of further pre-training the model on the dialog domain, followed by augmented data tuning. When comparing the data augmentation methods, we can see that the performance of the knowledge-grounded dialog augmentation method (GPT augmentation) is higher than the paraphrasing method (synonym replacement, filling mask, back-translation, paraphrase). This is due to the nature of knowledge-grounded dialogs, it is important to understand the context of the conversation, as well as to learn various knowledge forms so that relevant knowledge can be summarized and answered appropriately.

In the in-context learning results, the summary-clue generation method achieves the highest performance. Compared to clue-based generation, it

seems to generate responses that reflect the characteristics of the benchmark dataset well by including many few-shot exemplars. Also, it seems to generate better responses because it directly writes down the user’s intention rather than including the dialog as it is.

4.3.3 Human Evaluation Results

We gather three crowd workers to evaluate 50 randomly sampled data samples per model. We select four models that achieve good performance on the automatic metric. We evaluate three fine-tuning models trained with data augmented with back-translation, synonym replacement, and knowledge-grounded GPT augmentation, and one in-context model generated with summary-clue prompts. The evaluation criteria are accuracy, which measures how accurately the model reflects knowledge, and appropriateness, which measures how appropriate the response is as seen by a human, according to (Zhao et al., 2023). The results of the human evaluation are shown in Table 6. Unlike automatic evaluation, the preferences for in-context learning and fine-tuning are not significantly different. However, the model fine-tuned by including GPT-augmented data received the highest preference score from users. This confirms that our proposed approach, which appropriately uses LLM to learn in cooperation with a specific task model, is very effective. More details of the human evaluation can be found in appendix E.

5 Comparison with LLM Sub-task Solver

We have shown how to effectively use LLM to solve sub-tasks efficiently. By using a fine-tuning methodology through adapters and data augmentation methodology based on LLM’s knowledge, we are able to significantly increase the performance per task.

Our proposed methodology does not simply use LLM as a one-step sub-task solver, but utilizes LLM to improve the performance of the task solver model. To verify that our methodology outperforms LLM directly solving tasks, we compare the performance of our proposed methodology with that of LLM directly solving tasks. This is summarized in Table 7. In the KS task, similarity calculation is performed using embeddings from the *text-embedding-ada-002* model. As described in the table, we can see that utilizing LLM as in our proposed methodology is effective in solving SK-TOD problems.

	Accuracy	Appropriateness	Average
Back-translation	3.37	3.98	3.67
Synonym Replacement	3.42	3.78	3.60
GPT Augmentation	3.58	4.59	4.08
Summary-based Prompt	3.29	4.14	3.71

Table 6: Results of human evaluation.

Knowledge-Seeking Turn Detection					
	Precision	Recall	F1-score		
GPT-3.5-turbo	49.95	100	66.62		
Ours	99.54	99.93	99.73		
Knowledge Selection					
	Precision	F1-score	EMAcc		
Embedding search	40.90	44.43	0.82		
Ours	83.35	82.76	47.71		
Response Generation					
	BLEU	MT	R-1	R-2	R-L
GPT-3.5-turbo	6.78	17.18	34.14	12.02	24.87
Ours	10.35	19.01	37.91	15.42	29.37

Table 7: Comparison with LLM sub-task solver.

6 Conclusion

We propose an LLM-based framework for SK-TOD. In the KTD task, we fine-tune T5 using task-specific adapters. To address the challenge of relatively reduced data classified as knowledge-seeking turns in the KS and RG tasks, we employ knowledge-based data augmentation through LLM. Our three-stage data augmentation methodology involves last question generation, knowledge generation, and response generation for the RG task, as well as generating good and bad questions for the KS task. Experimental results show that our LLM-based framework achieves significantly higher performance compared to the baselines in the KS and RG tasks. Furthermore, our proposed framework for SK-TOD outperforms using LLM directly as a task solver, indicating its promising efficiency for SK-TOD tasks. Further research may be required to explore its efficiency further and understand its potential advantages more comprehensively. We plan to make the augmented data publicly available, hoping to contribute to the advancement of SK-TOD and future research in knowledge-based data augmentation.

Limitations

We based our entire framework on the LLM. We have tried to find the optimal prompts and made them publicly available. However, it is important to note that an LLM may generate different or unexpected responses when reproducing the experiment.

Ethics Statement

The augmented data was fully examined before publication. We filtered the data to ensure that it did not contain discrimination and hatred of any particular race, gender, region, or age. If we subsequently discover data that does contain bias, we will take immediate action to remove it.

Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2022-0-00621, Development of artificial intelligence technology that provides dialog-based multi-modal explainability)

References

- Namo Bang, Jeehyun Lee, and Myoung-Wan Koo. 2023. [Task-optimized adapters for an end-to-end task-oriented dialogue system](#). In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 7355–7369. Association for Computational Linguistics.
- Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. Inpars: Data augmentation for information retrieval using large language models. *arXiv preprint arXiv:2202.05144*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Jishnu Ray Chowdhury, Yong Zhuang, and Shuyi Wang. 2022. [Novelty controlled paraphrase generation with retrieval augmented conditional prompt tuning](#). In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 10535–10544. AAAI Press.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.
- Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Lichao Sun, Quanzheng Li, Dinggang Shen, Tianming Liu, and Xiang Li. 2023. [Auggpt: Leveraging chatgpt for text data augmentation](#).
- Prithviraj Damodaran. 2021. Parrot: Paraphrase generation for nlu.
- Jun Gao, Changlong Yu, Wei Wang, Huan Zhao, and Ruifeng Xu. 2023. [Mask-then-fill: A flexible and effective data augmentation framework for event extraction](#). *CoRR*, abs/2301.02427.
- Silin Gao, Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. [Paraphrase augmented task-oriented dialog generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 639–649. Association for Computational Linguistics.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. [Parameter-efficient transfer learning for NLP](#). In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.
- Seokhwan Kim, Yang Liu, Di Jin, Alexandros Papanagelis, Behnam Hedayatnia, Karthik Gopalakrishnan, and Dilek Hakkani-Tür. 2022. Knowledge-grounded task-oriented dialogue modeling on spoken conversations track at dstc10.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Bing Liu and Ian Lane. 2017. An end-to-end trainable neural network model with belief tracking for task-oriented dialog. *arXiv preprint arXiv:1708.05956*.
- Shikib Mehri, Evgeniia Razumovskaia, Tiancheng Zhao, and Maxine Eskenazi. 2019. [Pretraining methods for dialog context representation learning](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3836–3845, Florence, Italy. Association for Computational Linguistics.
- Tong Niu, Semih Yavuz, Yingbo Zhou, Nitish Shirish Keskar, Huan Wang, and Caiming Xiong. 2021. [Unsupervised paraphrasing with pretrained language models](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5136–5150. Association for Computational Linguistics.

OpenAI. 2023. [GPT-4 Technical Report](#). *arXiv e-prints*, page arXiv:2303.08774.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Improving neural machine translation models with monolingual data](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Noam Shazeer and Mitchell Stern. 2018. [Adafactor: Adaptive learning rates with sublinear memory cost](#). In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR.

Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2021. Multi-task pre-training for plug-and-play task-oriented dialogue system. *arXiv preprint arXiv:2109.14739*.

Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2022. [Multi-task pre-training for plug-and-play task-oriented dialogue system](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4661–4676, Dublin, Ireland. Association for Computational Linguistics.

Xin Tian, Yingzhan Lin, Mengfei Song, Siqi Bao, Fan Wang, Huang He, Shuqi Sun, and Hua Wu. 2022. [Q-tod: A query-driven task-oriented dialogue system](#).

Peifeng Wang, Aaron Chan, Filip Ilievski, Muhao Chen, and Xiang Ren. 2022. Pinto: Faithful language reasoning using prompt-generated rationales. *arXiv preprint arXiv:2211.01562*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. [Self-instruct: Aligning language models with self-generated instructions](#).

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei and Kai Zou. 2019. [Eda: Easy data augmentation techniques for boosting performance on text classification tasks](#).

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. [PEGASUS: pre-training with extracted gap-sentences for abstractive summarization](#). In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Qiang Zhang, Jason Naradowsky, and Yusuke Miyao. 2023. Ask an expert: Leveraging language models to improve strategic reasoning in goal-oriented dialogue models. *arXiv preprint arXiv:2305.17878*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

Chao Zhao, Spandana Gella, Seokhwan Kim, Di Jin, Devamanyu Hazarika, Alexandros Papangelis, Behnam Hedayatnia, Mahdi Namazifar, Yang Liu, and Dilek Hakkani-Tur. 2023. ["what do others think?": Task-oriented conversational modeling with subjective knowledge](#).

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive dialogue state tracker. *arXiv preprint arXiv:1805.09655*.

Appendices

A How to Build Version 6 data for Knowledge Selection Task

This method involves generating a history of previous system and user utterances for various single utterances generated through data augmentation techniques. Given the last user utterance, previous system and user utterances are generated sequentially. 2-shot exemplars are provided. A detailed prompt example looks like the Figure 4.

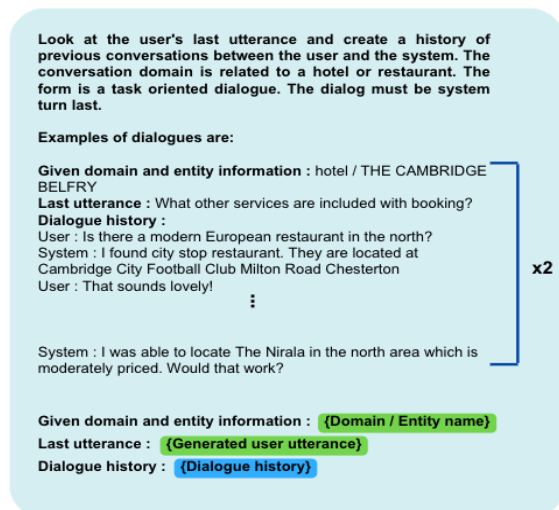


Figure 4: Example of history generation prompts for building version 6 data in the KS task. The bold text represents the default prompt format, and the input provided is highlighted in green. The blue text denotes the portion generated by the model.

B Details of Flan-T5-XL Instruction Template

As a preliminary step for Flan-T5 fine-tuning, we conducted an experiment to determine which in-

	BLEU	MT	R-1	R-2	R-L
(a) instruction	9.61	17.15	35.72	14.67	27.98
(b) instruction	9.17	16.82	34.02	13.28	25.88

Table 8: Comparison of Flan-T5-XL instruction templates.

instruction format is most effective for the instructions in Flan-T5 prompts. We compared the form (a) "system: <extra_id_0>", which contains special token, which is mainly used to format the input and output text of T5, and the form (b) "generate system response based on knowledge and dialog context: ", which describes the task to be performed by the model in natural language (Tian et al., 2022). The results of fine-tuning using each instruction and comparing the performance are summarized in table 8. As a result, the special token form (a) outperformed the natural language form (b). Since the Flan model is instruction tuned by focusing on the model’s understanding of natural language instructions, we expected that the performance would be higher when using natural language instructions, but the performance was higher when using special token. Therefore, we unified the instructions in all fine-tuning experiments to (a) with special token. Also, Table 9 shows examples of each instruction template.

C Context-based Data Augmentation Methods for Response Generation Task

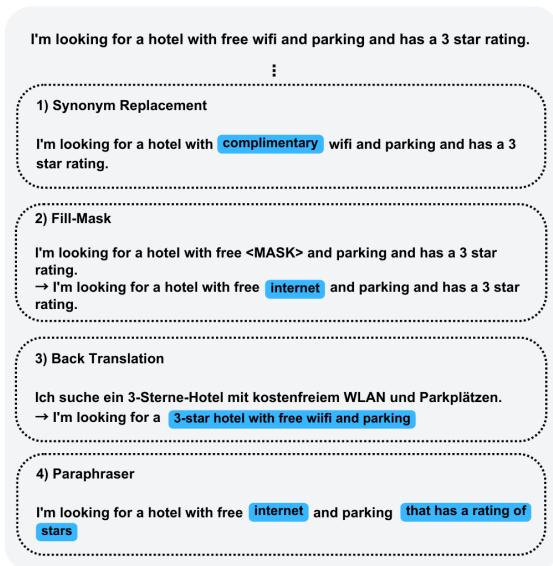


Figure 5: Examples of inpaint methods without knowledge for the RG task.

When fine-tuning the Flan-T5-XL model, we used context-based data augmentation to increase performance. The augmentation methods without using Knowledge are shown in the figure 5.

D In-context Learning Prompt Examples

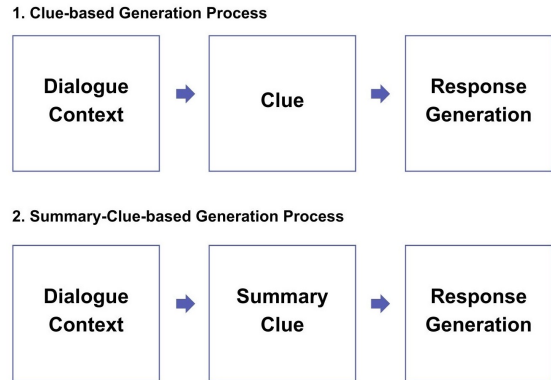


Figure 6: Clue-based Generation Prompting.

Manual prompt, contains knowledge and dialog context in the form of a python dictionary data structure. It also organizes the prompt by putting the user’s question at the end and the instruction to answer it at the end. The performance presented in the paper is the result of a 2-shot prompt. Table 10 shows the examples.

Clue-based generation prompting is a way to have GPT generate the clues it needs to generate an answer, and then generate a response based on those clues. In the two-stage generation process, the GPT initially generates three clues: 1) what the user wants, 2) what information it needs to provide to the user, and 3) what it can ask the user back. An example prompt might look like this. The performance presented in the paper is based on a 4-shot prompt. Figure 6 shows the process and table 11 shows the examples.

Summary-clue generation is a two-stage generation process where GPT summarizes the context of the conversation and creates a summary-clue of what the user wants. The ability to include more few-shot exemplars improves performance by showing more example answers that can be used to generate answers that fit the automatic metric. Figure 6 shows the process and table 12 shows the examples.

E Human Evaluation

We asked crowd workers to evaluate each response generated from different models according to the

Example of (a) Instruction
<p>From domain hotel and entity BRIDGE GUEST HOUSE, knowledge1: The room was clean and comfortable and not expensive. knowledge2: It could ruin your stay if you mind that kind of thing. knowledge3: Sadly though, I found that the bed in the room wasn't very comfortable at all. knowledge4: I do have to say, though, the bed is extremely uncomfortable. knowledge5: and the interior of the room was very good and bed was also very much comfortable.</p> <p>user:Can you help me find a place to stay that is moderately priced and includes free wifi? system:sure, i have 17 options for you user:Are any of them in the south? I'd like free parking too. system:Yes, two are in the south and both have free parking and internet. I recommend the Bridge Guesthouse. Would you like me to book a reservation? user:I have back issues. Does this place have comfortable beds?</p> <p>system: <extra_id_0></p>
Example of (b) Instruction
<p>From domain hotel and entity BRIDGE GUEST HOUSE, knowledge1: The room was clean and comfortable and not expensive. knowledge2: It could ruin your stay if you mind that kind of thing. knowledge3: Sadly though, I found that the bed in the room wasn't very comfortable at all. knowledge4: I do have to say, though, the bed is extremely uncomfortable. knowledge5: and the interior of the room was very good and bed was also very much comfortable.</p> <p>user:Can you help me find a place to stay that is moderately priced and includes free wifi? system:sure, i have 17 options for you user:Are any of them in the south? I'd like free parking too. system:Yes, two are in the south and both have free parking and internet. I recommend the Bridge Guesthouse. Would you like me to book a reservation? user:I have back issues. Does this place have comfortable beds?</p> <p>generate system response based on knowledge and dialog context:</p>

Table 9: Examples of T5 instruction templates.

following criteria:

- Appropriateness: how well the response is naturally connected to the conversation
- Accuracy: how accurate a system agent's response reflects the opinions and sentiments

This criteria is based on DSTC9 Track1 and DSTC10 Track2 human evaluation processes. Figure 7 shows the details of human evaluation instruction which was given as a guideline to crowd workers.

Manual Prompt Example

##

{"reviews of HOBSONS HOUSE":

["I also saw some hairs in the bathroom, which was evidently not well cleaned.",

"Upon using the bathroom facilities, we all saw that they were pristine and it looks like they had been recently cleaned.",

"The view from the room was also amazing, and the cleanliness of the bathroom was top notch."],

"dialog context": ["U: Can you help me find guesthouses in the west of town?",

"S: Of course, I have two available one is cheap and one is moderate what do you prefer?",

"U: Do either of them have a 3 star rating?",

"S: Yes, the Hobsons House has 3 stars and meets your criteria.

Would you like me to make a reservation for you?"]}]

answer as a system based on reviews and FAQs

U: Are their bathrooms adequately sanitized?

S: Yes, most of the guests at Hobsons House have rated the bathrooms very high on cleanliness, although there is at least one report of stray hairs in the bathroom.

##

{"reviews of THE CAMBRIDGE BELFRY":

["One of the best things about the hotel is the view of the mountains from the balcony.",

"The staff was polite they took out luggage to our room, and our room had a spectacular view of the forest and nature."],

"dialog context": ["U: I'm looking for information on the cambridge belfry hotel.",

"S: the cambridge belfry is cheap, located in the west, and has 4 stars. It has free internet and parking."}]

answer as a system based on reviews and FAQs

U: Do you know anything about the views from the rooms as I hope to have a nice view?

S: Yes, the rooms at the Cambridge Belfry have very nice views.

Can I answer anything else, or would you like to book it?

##

{"reviews of ASHLEY HOTEL":

["Looking out of the window was a great view, and we loved it.",

"There was a nice size refrigerator and a beautiful view out the window of the 7th floor."],

"dialog context": ["U: I'm looking to stay at a 3 star hotel in the north.",

"S: Sorry, I have no results for that query. Would you like to try a different area of town?",

"U: Are there any moderate priced hotels in the North?",

"S: Yes I have two. Would you like me to book one?",

"U: I need a hotel to include free parking; does either have that?",

"S: Yes both of them have free parking.",

"U: Which one would you recommend?",

"S: How about the Ashley hotel?",

"U: Is the Ashley hotel a 3 star hotel?",

"S: the ashley is actually a 2 star hotel."}]

answer as a system based on reviews and FAQs

U: Does this hotel have rooms with a good view of the neighborhood?

S:

Table 10: Examples of manual prompt.

Clue-based Generation Example	
<pre> {"dialog context": ["U: I'm looking to stay at a 3 star hotel in the north.", "S: Sorry, I have no results for that query. Would you like to try a different area of town?", "U: Are there any moderate priced hotels in the North?", "S: Yes I have two. Would you like me to book one?", "U: I need a hotel to include free parking; does either have that?", "S: Yes both of them have free parking.", "U: Which one would you recommend?", "S: How about the Ashley hotel?", "U: Is the Ashley hotel a 3 star hotel?", "S: the ashley is actually a 2 star hotel.", "U: Does this hotel have rooms with a good view of the neighborhood?"]} </pre> <p>based on dialog context, figure out</p> <ol style="list-style-type: none"> 1) what the user wants 2) what information you should provide to answer user's last question 3) what can you ask user? <p>write in one to two sentences.</p>	<p>role: you are a chatbot assistant system.</p> <pre> {"reviews of ASHLEY HOTEL": ["There was a nice size refrigerator and a beautiful view out the window of the 7th floor.", "Looking out of the window was a great view, and we loved it."],} </pre> <p>based on FAQs or reviews, answer the question</p> <p>"U: Does this hotel have rooms with a good view of the neighborhood?"</p> <p>[clues]</p> <ol style="list-style-type: none"> 1) The user wants a moderate priced hotel in the north with free parking and rooms with a good view of the neighborhood. (basic) 2) To answer the user's last question, you should provide information about whether the Ashley hotel has rooms with a good view of the neighborhood. (important) 3) You can ask the user if they would like you to check the availability of rooms with a good view at the Ashley hotel or if they have any other preferences regarding the hotel. (optional) <p>answer in 32 tokens.</p>

Table 11: Examples of clue-based generation prompt. First row shows the input of GPT to generate clues. Second row shows the input of GPT to generate responses based on knowledge and clues.

Summary-clue Generation Example
<p>{ "dialog context": ["U: I am looking for a place to dine that has expensive romanian food.", "S: No I'm sorry. Something else perhaps?", "U: Is there one that serves Chinese instead?", "S: I'm showing 9 entries throughout the area. Do you have a preference between center, north, east, or south?", "U: I have no preference. I just need to book it for 8 at 12:45 on Thursday.", "S: I have you booked at Yu Garden. The table will be reserved for 15 minutes. Your reference number is: D6LEPNCC. Anything else I can do for you?", "U: Do you know if the restaurant has a good outdoor dining space?"]}</p> <p>summarize dialog context and what the user wants, with a focus on the last question. write in one to two sentences.</p>
<p>role: you are a chatbot assistant system. situation: The user is asking if the restaurant they have booked, Yu Garden, has a good outdoor dining space.</p> <p>{ "FAQs of YU GARDEN": ["YU GARDEN does offer outdoor seating."], "reviews of YU GARDEN": ["When I first arrived I noticed out front there was a nice place for outdoor dining so I asked to be seated there.", "However, the view from our table was facing a wall and cramped wish they would have sat us outside."]}</p> <p>based on FAQs or reviews, answer the user's question: Do you know if the restaurant has a good outdoor dining space? you should answer in 32 tokens.</p>

Table 12: Examples of summary-clue generation prompt. First row shows the input of GPT to generate clues. Second row shows the input of GPT to generate responses based on knowledge and situations.

Human Evaluation Instruction

We provide a conversation between a user and a system agent which ends with a user input, and the entire user reviews that are relevant to the user's input.

Please first read a conversation and relevant reviews(knowledges), and score the generated responses on a scale of 1 - 5 on the following criteria:

- **Appropriateness:** Appropriateness means how well the response is naturally connected to the conversation. A score of 1 means that the response is very inappropriate and it is not naturally connected to the conversation. A score of 5 means that the response is very appropriate and it is very naturally connected to the conversation.
- **Accuracy:** Accuracy means how accurate a system agent's response reflects the opinions and sentiments that you obtained from the reviews. A score of 1 means that the response is completely wrong. A score of 5 means that the response is completely accurate against the sentiment distribution of given knowledge snippets.

Figure 7: Example of human evaluation instruction.