Prompting, Retrieval, Training: An exploration of different approaches for task-oriented dialogue generation

Gonçalo Raposo **Luísa Coheur** **Bruno Martins**
INESC-ID, Instituto Superior Técnico, Universidade de Lisboa
{goncalo.cascalho.raposo, luisa.coheur, bruno.g.martins}@tecnico.ulisboa.pt

Abstract

Task-oriented dialogue systems need to generate appropriate responses to help fulfill users' requests. This paper explores different strategies, namely prompting, retrieval, and fine-tuning, for task-oriented dialogue generation. Through a systematic evaluation, we aim to provide valuable insights and guidelines for researchers and practitioners working on developing efficient and effective dialogue systems for real-world applications. Evaluation is performed on the MultiWOZ and Taskmaster-2 datasets, and we test various versions of FLAN-T5, GPT-3.5, and GPT-4 models. Costs associated with running these models are analyzed, and dialogue evaluation is briefly discussed. Our findings suggest that when testing data differs from the training data, fine-tuning may decrease performance, favoring a combination of a more general language model and a prompting mechanism based on retrieved examples.

1 Introduction

Task-oriented dialogue systems need to generate appropriate responses to help fulfill users' requests. Recent advancements in Natural Language Processing (NLP) have produced a shift towards leveraging large pretrained language models to tackle the generation challenge (Zhang et al., 2020). By prompting these models with a few examples, their performance has been shown to surpass traditional approaches, eliminating the need for extensive model training (Brown et al., 2020; Zhang et al., 2022).

In this paper, we explore different approaches for task-oriented dialogue generation, namely through the use of prompting, retrieval mechanisms, and fine-tuning. We investigate the best strategies to leverage these approaches, considering the integration of past conversation information, the selection of appropriate retrieval methods, and the assessment of the benefits of fine-tuning (Roller et al., 2021; Izacard et al., 2022; Peng et al., 2022).

During our investigation, we assessed various state-of-the-art instruction-based models, including different size versions of FLAN-T5 (Chung et al., 2022), GPT-3.5, and GPT-4, provided by OpenAI (Ouyang et al., 2022; OpenAI, 2023). These models, known for their impressive language generation capabilities, serve as the foundation for our experiments, through which we tested different strategies. We evaluate the performance of these models on widely used benchmark datasets, namely MultiWOZ and Taskmaster-2, which offer diverse and challenging dialogue scenarios (Zang et al., 2020; Byrne et al., 2019). Additionally, we analyze the computational costs associated with running the models, considering the trade-off between performance and resource requirements. Moreover, we discuss dialogue system evaluation, addressing the metrics and criteria that best capture the quality and effectiveness of task-oriented dialogue generation (Sellam et al., 2020; Nekvinda and Dušek, 2021).

The main contributions of this paper are[1]:

- Investigate different approaches for task-oriented dialogue generation, including prompting, use of retrieval mechanisms, and fine-tuning.
- Advocate for the combination of a large pretrained language model with the proposed retrieval mechanism when the testing data significantly deviates from the training data, showcasing its effectiveness and cost-efficiency.
- Examine the positioning of GPT-3.5 and GPT-4 models, comparing them with both pretrained and fine-tuned models, to understand their performance characteristics, advantages, and costs.

2 Related work

Task-oriented dialogue generation has garnered significant attention, leading to a wide range of research efforts. Recent studies have focused on the

[1]We make all of our code available online at https://github.com/gonced8/dialogue-retrieval

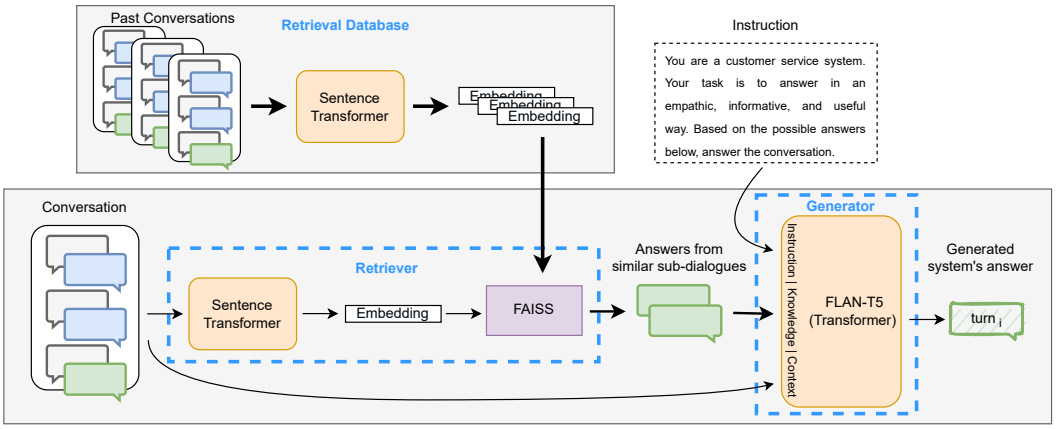

Figure 1: Our main approach for answer generation enhanced with possible answers retrieved from past conversations. During inference, our system starts by retrieving the possible answers and then includes them in the prompt given to the FLAN-T5 model, used to generate the system's response in the context of a dialogue.

use of large pretrained language models for dialogue systems. Radford et al. (2018) introduced GPT, a Transformer (Vaswani et al., 2017) using generative pretraining, which demonstrated impressive performance in various NLP tasks. Subsequent research explored the benefits of fine-tuning pretrained models specifically for dialogue generation tasks. For instance, Lin et al. (2020) proposed MinTL, a system that fine-tuned a pretrained model on task-oriented data and established new state-of-the-art results. Similarly, Thoppilan et al. (2022) employed fine-tuning on a larger pretrained model of approximately 37 B parameters and used around 1.56 T words of public dialogue data and web text, improving in all metrics.

Prompting has emerged as a valuable technique for improving the performance of pretrained language models. It involves providing specific examples or instructions as input to guide the generation process. Brown et al. (2020) demonstrated the effectiveness of prompts when using language models to generate coherent and contextually appropriate responses. A recent work by Gupta et al. (2022) addresses prompting in the context of dialogue systems, showing how instruction tuning may benefit certain test tasks.

Retrieval-enhanced methods have also been extensively explored in dialogue systems. Yang et al. (2019) integrated text retrieval and text generation models to build a hybrid conversational system that outperformed retrieval-based and generation-based approaches. In addition, several studies have also incorporated retrieval mechanisms in combination with generative models to enhance dialogue system performance (Roller et al., 2021; Shuster et al., 2022; Thoppilan et al., 2022).

While the aforementioned studies have made substantial contributions to the field, this paper aims to expand upon the existing literature by thoroughly investigating the integration of prompting, retrieval mechanisms, and fine-tuning in task-oriented dialogue generation. Specifically, we explore the efficacy of these approaches and analyze their impact on system performance, considering both the quality of generated responses and the computational costs associated with running the models. Furthermore, as far as we know, we are the first work employing the GPT-3.5 and GPT-4 models for the MultiWOZ and Taskmaster-2 datasets, establishing baselines for each.

3 Method

In our main approach, we propose to use a dense retrieval model that, given a dialogue, will retrieve other similar dialogues. We then use their answers to generate a new answer using a Transformer. Figure 1 illustrates how our system can be used for inference, depicting its components.

3.1 Dense retrieval of dialogue answers

We use dense retrieval (Karpukhin et al., 2020; Gao et al., 2023) to obtain relevant responses given a conversation context. Since the task of retrieving responses for dialogues is not necessarily equivalent to document or passage retrieval (Penha and Hauff, 2023), we considered two possible approaches: (1) Encode the current conversation context and compare it to a database of encoded past contexts. The returned relevant responses will correspond to the turns immediately after each of the indexed contexts; (2) Encode the current conversation context and query a database of encoded past

responses. The returned relevant responses will be those whose embeddings are the most similar to the query/context embedding.

The library Sentence-Transformers (Reimers and Gurevych, 2019) provides models already pretrained for tasks like text clustering or semantic search, that can be used to perform the described response retrieval. In particular, it provides Transformer-based encoders that can be used to compute text embeddings, and then compare the embeddings with a similarity function (e.g., cosine-similarity or dot product).

To implement the two approaches described, we considered two of the top pretrained models provided by Sentence-Transformers: `all-mpnet-base-v2` and `multi-qa-mpnet-base-dot-v1`. These models are both fine-tuned versions of the pretrained MPNet model (Song et al., 2020) using a contrastive loss. In particular, `all-mpnet-base-v2` was fine-tuned to be used for information retrieval, clustering, or sentence similarity tasks, making it more appropriate for our first approach. On the other hand, `multi-qa-mpnet-base-dot-v1` was fine-tuned for semantic search and it is intended to be used to pair queries/questions with relevant text paragraphs. Thus, we used it for our second approach, given the size difference between contexts and responses. For both models, we use dot product as our similarity function.

Using a conversation context as a query differs significantly from relevant passage retrieval. Some studies perform question rewriting to circumvent this issue and use a rewritten context-independent version of the last turn as the query (Raposo et al., 2022). Since question rewriting may also require additional training, we simply fine-tuned the retrieval encoder for conversational text.

We specifically used weakly supervised learning to train our encoder. Starting from an unlabeled dataset of conversations, we made sets of queries (conversation contexts) and documents (either 1. conversation contexts or 2. conversation responses). Then, given a random batch of query embeddings, we compute the similarity with the document embeddings. With the option 2., it makes sense to match the context to the corresponding response. However, with option 1., we match the context with a different context from that batch based on the similarity between responses (measured using ROUGE). We train the encoder using a cross-entropy loss (Wang et al., 2020):

$$\mathrm{L} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp\left(Q_{N\times d}\cdot D_{N\times d}^{T}\right)_{i,k}}{\sum_{j=1}^{N}\exp\left(Q_{N\times d}\cdot D_{N\times d}^{T}\right)_{i,j}} \quad (1)$$

where $Q_{N\times d}$ is a matrix composed by N queries embeddings of size d, $D_{N\times d}$ a matrix composed by the corresponding N document embeddings of size d. The index k will correspond to the target document. The similarity computation uses the dot product, and no temperature parameter is applied.

3.2 Answer generation

We use the pretrained Transformer named FLAN-T5 (Chung et al., 2022), which is an enhanced version of the T5 model (Raffel et al., 2020) that was fine-tuned using instructions and is reported to achieve strong few-shot performance.

3.2.1 Generation-only approach

We start by evaluating FLAN-T5 in a zero-shot setting, using no examples of possible answers. In practice, our approach consisted in giving the model the following prompt:

```
You are a customer service system. Your
task is to answer in an empathic,
informative, and useful way. Answer the
conversation.
Conversation:
{conversation context}
```

This prompt is followed by the conversation context and the model generates the response.

3.2.2 Generating based on past answers

To incorporate the information from the retrieved past answers, we simply concatenate them in the input that is given to the generation model. This approach is similar to the work by Ram et al. (2023) and its main benefits are its simplicity and versatility, which allow it to be implemented with any generative model. Thus, FLAN-T5 is used in a few-shot setting with the following prompt:

```
You are a customer service system. Your
task is to answer in an empathic,
informative, and useful way. Based on
the possible answers below, answer the
conversation.
Possible answers:
{possible answers}
Conversation:
{conversation context}
```

3.2.3 Fine-tuning for answer generation

We described how we used retrieved past answers as examples for our generation model, which related works have shown to improve performance. In addition, we also study how fine-tuning the same model affects the achieved performance. Using the same prompts mentioned above, we train our models in both scenarios: with and without retrieval. During training and evaluation, we are careful to avoid data leakage in the retrieved answers (e.g., we index the training dataset, and we do not retrieve responses from the same conversation).

3.2.4 Open-AI models

Given the recent popularity and impressive performance of OpenAI's large language models – ChatGPT and GPT-4 – we also performed some experiments using their API. Similarly to the FLAN-T5 model, these chat-based models were also fine-tuned in an instruction-following setting but using Reinforcement Learning with Human Feedback (RLHF) for optimization (Ouyang et al., 2022; OpenAI, 2023). For reproducibility, we reused the same prompts from FLAN-T5 and evaluated both the zero-shot and few-shot settings.

4 Experimental setup

Broadly, our experiments consisted in testing different generation models on the task of answer generation in task-oriented dialogues. In some cases, this also involved the use of information retrieval mechanisms or fine-tuning models.

4.1 Implementation details

Regarding dense retrieval, we use the models from Sentence Transformers to compute the embeddings, together with FAISS (Johnson et al., 2019) to index and search them. When training the retrieval modules, we used a batch size of 64 samples and the AdamW optimizer (Loshchilov and Hutter, 2019).

For FLAN-T5, we use the checkpoints available on Hugging Face (Wolf et al., 2020). In particular, we use the small, large, and XL versions. To test and train the models, we use the Transformers library from Hugging Face along with the PyTorch framework (Paszke et al., 2019). We use the AdamW optimizer and train our models for a maximum of 20 epochs with patience of 5 steps. The batch size varied for each model due to limitations on GPU memory, but the effective batch size was kept at 64 samples. All our local models were trained and tested using a `NVIDIA Quadro RTX 6000` GPU with 24 GB of memory. As for the OpenAI models, we use their API through the provided Python package, keeping the default settings.

4.2 Task-oriented datasets

Starting from a task-oriented dataset, we extract a dataset consisting of sub-dialogues. Based on Nekvinda and Dušek (2022), we chose to use a maximum of 6 turns for each sub-dialogue, which seemed like a good compromise between providing enough context but not too long. The extracted sub-dialogues can be obtained by sliding a window of size 6 turns over the original dialogue, with a stride of 2 turns to always end in a system's turn. Depending on the speaker, we prepend each turn with "User: " or "System: ".

We apply this technique to the MultiWOZ 2.2 and Taskmaster-2 task-oriented datasets. As Taskmaster-2 has not already predefined dataset splits, we randomly select 1k dialogues for both validation and testing, ensuring a balanced distribution across domains. Table 1 shows a summary of the sizes of the obtained datasets of sub-dialogues.

Table 1: Number of samples for each dataset split after applying the preprocessing that consists of splitting each dialogue into multiple sub-dialogues.

Dataset	Train	Validation	Test
MultiWOZ 2.2	56776	7374	7372
Taskmaster-2	120892	7997	8038

4.3 Automatic evaluation metrics

To measure the performance of our models, we compare the returned answers to the ground truth answers. In particular, we use automatic metrics based on lexical similarity (i.e., BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004)) and on semantics similarity (i.e., BERTScore (Zhang* et al., 2020) and BLEURT (Sellam et al., 2020)). Additionally, we score the quality of the generated answers using QualityAdapt, a reference-free metric that achieves state-of-the-art performance on overall dialogue quality estimation through adapter fusion (Mendonca et al., 2022).

5 Results

5.1 Retrieval-only responses

The first approach to obtain the dialogue response that we evaluated consists of using a retrieval-only model. Given a conversation context as a query, its

Table 2: Performance of using only a retrieval model to return the response. Two pretrained models are compared to their fine-tuned versions on MultiWOZ. The models differ in how they perform retrieval: indexing the contexts and returning the next response, against indexing the responses.

Retrieval Model	query-document	BLEU	ROUGEL-F1	BERTScore	BLEURT	QualityAdapt
all-mpnet-base-v2	context-context	0.0652	0.1767	0.2032	0.4022	0.8255
multi-qa-mpnet-base-dot-v1	context-answer	0.0270	0.1456	0.1382	0.3700	0.9141
Fine-tuned all-mpnet-base-v2	context-context	**0.0940**	**0.2622**	**0.3169**	**0.4762**	0.8905
Fine-tuned multi-qa-mpnet-base-dot-v1	context-answer	0.0759	0.2406	0.3030	0.4633	**0.9317**

Table 3: Performance of using only the generation model to generate the response (zero-shot). We use pretrained FLAN-T5 models and fine-tuned versions. FLAN-T5 XL was not fine-tuned due to the large GPU memory required.

Generation Model	BLEU	ROUGEL-F1	BERTScore	BLEURT	QualityAdapt
FLAN-T5 (small)	0.0234	0.1200	0.0967	0.3374	0.8362
FLAN-T5 (large)	0.0400	0.1456	0.1164	0.3840	0.9090
FLAN-T5 (XL)	0.0367	0.1400	0.1389	0.3593	0.9131
Fine-tuned FLAN-T5 (small)	0.1231	0.2764	**0.3236**	0.4843	**0.9474**
Fine-tuned FLAN-T5 (large)	**0.1255**	**0.2795**	0.3079	**0.4925**	0.9433

task is to retrieve the corresponding answer from a database. We evaluate indexing past conversation contexts and indexing only past answers, as described in Subsection 3.1. During the evaluation, we used the conversations from the training dataset as the aforementioned past conversations.

In Table 2 we report the results obtained by using two different pretrained models from Sentence-Transformers. The approach that indexed the contexts (matching contexts to similar contexts) obtained much better results. The lower performance of the model that indexed the answers can be explained by the mismatch of the pretraining objective and the current task: matching questions to relevant passages is different from matching answers to conversation contexts. After fine-tuning each of the retrieval models, the performance increased in both cases and it became closer, although the context-context approach remained better overall.

5.2 Generation-only responses

The second approach we tested consists of using the language model FLAN-T5 in a zero-shot setting (with no examples, only the conversation context). Given the maximum input length of the model of 512 tokens, we filtered overflowing samples. During decoding, we initialize each generation with "`System: `" and decode using beam search (`n_beams = 4`), since this showed more consistent results than other sampling methods.

In Table 3, we report the results obtained with three variations of FLAN-T5. When comparing the pretrained versions without fine-tuning, the large and XL versions, as expected, showed better results than the small version. However, analyzing only the automatic metrics, it is not evident that XL is better than the large version. Compared to the retrieval-only results (Table 2), the generation-only approach is only better after fine-tuning.

5.3 Retrieval-enhanced generation

As described, we explore combining retrieved answers with the generation model. We retrieve the top-5 possible answers and add them to the prompt of FLAN-T5. The objective is for the model to generate an answer similar to those retrieved.

5.3.1 Indexing contexts or answers

While the results presented in Table 2 indicate superior performance when indexing the conversation contexts, we conducted a comparative analysis by indexing the answers. As we show in Table 4, when combined with the generation model, the method that indexed the answers actually obtained slightly better results. Although the performance is not notably higher, indexing the answers is also computationally lighter than indexing the contexts, due to the smaller sequence size. Hence, we chose to index the answers as our preferred retrieval approach.

Moreover, compared to the zero-shot results in Table 3, introducing the retrieved answers increases the performance, almost doubling some of the automatic metrics. Nonetheless, the fine-tuned version of FLAN-T5 is still better than this few-shot approach (with retrieval but without fine-tuning).

5.3.2 Fine-tuning generation with retrieval

Since both fine-tuning and retrieval showed increased scores in the automatic metrics, our next experiment consisted in fine-tuning the generation

Table 4: Performance of different approaches using the retrieval model paired with a generation model. We compare indexing the contexts against indexing the answers. The retrieval models are the fine-tuned versions, and the FLAN-T5 models are the pretrained versions.

Generation Model	Retrieval	BLEU	ROUGEL-F1	BERTScore	BLEURT	QualityAdapt
FLAN-T5 (small)	context-context	0.0354	0.1160	0.1274	0.3212	0.8061
FLAN-T5 (large)		0.0637	0.1775	0.1780	0.3944	0.8905
FLAN-T5 (XL)		0.0644	0.1966	0.2068	0.4144	0.9064
FLAN-T5 (small)	context-answer	0.0445	0.1261	0.0516	0.3335	0.8061
FLAN-T5 (large)		0.0683	0.1804	0.1970	0.4036	0.8976
FLAN-T5 (XL)		**0.0693**	**0.2056**	**0.2327**	**0.4240**	**0.9217**

Table 5: Performance of using both retrieval and generation to obtain the response. The FLAN-T5 models were fine-tuned with/without using retrieved answers. The retrieval model is the fine-tuned version of indexing answers.

Model	Retrieval	BLEU	ROUGEL-F1	BERTScore	BLEURT	QualityAdapt
Fine-tuned FLAN-T5 (small)	w/o retrieval	0.1231	0.2764	0.3236	0.4843	**0.9474**
Fine-tuned FLAN-T5 (large)		0.1255	0.2795	0.3079	0.4925	0.9433
Fine-tuned FLAN-T5 (small)	w/ retrieval	0.1307	0.2938	0.3268	0.5015	0.9405
Fine-tuned FLAN-T5 (large)		**0.1374**	**0.2976**	**0.3359**	**0.5033**	0.9443

model with the retrieved candidates. Our aim was for the FLAN-T5 model to learn to make better use of the retrieved answers during generation.

Table 5 shows the results of our fine-tuned FLAN-T5 models with and without retrieval. Although combining fine-tuning and retrieval resulted in higher scores in terms of automatic metrics, the small increment suggests that most of the performance gain results from fine-tuning and not much from the additional retrieved information.

5.4 Adapting to a different dataset

The results reported until now suggested that fine-tuning a generation model with the retrieved answers is the best approach in our evaluation with MultiWOZ. However, one of the downsides of fine-tuning these large language models is that they might lose some of their generalization capabilities. Suppose you want to deploy a dialogue system for a customer service application and still do not have enough data to fine-tune your models for the specific type of data it will see. To obtain a better insight on what is the best approach in terms of fine-tuning and retrieval, we also evaluate how our system adapted to a different task-oriented dataset.

In Table 6 we report the results of FLAN-T5 large in the Taskmaster-2 dataset. In the first two rows, we show the results obtained using only pretrained models. As the results suggest, prompting the generation model with possible answers obtained using an out-of-the-box pretrained retriever even tends to decrease its performance. We posit that without fine-tuning, the retrieval model struggles with conversational text (e.g., it does not focus on the last turn) and ends up introducing answers that are not very similar to the ground truth response.

As for the results obtained with fine-tuned models, the most effective approach seems to be using only a fine-tuned version of the retrieval model paired with the pretrained version of the generation model. In particular, when we only used a generation model fine-tuned in MultiWOZ (without retrieval) the results were even worse than without fine-tuning. This suggests that, although the format and structure of the data were similar (task-oriented dialogues), the fine-tuned model ended up being too fine-tuned to the content style of MultiWOZ, performing poorly in Taskmaster-2.

5.5 Comparing to GPT-3.5-turbo and GPT-4

Although these models allow for a larger input size, we considered the top-5 retrieved answers in the few-shot experiments.

In Table 7, we report the results obtained in the MultiWOZ dataset with our best model and those obtained with the models GPT-3.5-turbo and GPT-4. As expected, both OpenAI models showed a better performance when augmented with retrieved answers in a few-shot setting. Compared to our previous zero-shot results in Table 3, GPT-3.5-turbo and GPT-4 are better than a pretrained FLAN-T5 but slightly inferior to a fine-tuned version. The same can be said when considering Tables 4 and 5. In essence, combining and fine-tuning both the retrieval and generation models on data similar to the one seen during inference achieved better per-

Table 6: Evaluation of how the fine-tuned models (retrieval and generation) adapt to a different dataset – Taskmaster. The generation model used was FLAN-T5-Large. We compared using pretrained models in zero-shot and few-shot (with retrieval) settings, against fine-tuning some of the modules on MultiWOZ.

Method	BLEU	ROUGEL-F1	BERTScore	BLEURT	QualityAdapt
zero-shot	0.0368	0.1263	0.2178	**0.3960**	**0.8861**
with retrieval	0.0375	0.1100	0.2087	0.3626	0.8518
fine-tuned generation	0.0194	0.1115	0.1739	0.3801	0.8637
fine-tuned retrieval	**0.0441**	0.1226	**0.2368**	0.3710	0.8669
all fine-tuned	0.0266	**0.1284**	0.1932	0.3859	0.8714

Table 7: Evaluation using pretrained large language models from OpenAI (GPT-3.5-turbo and GPT-4) on MultiWOZ. We compare using no examples (zero-shot) against prompting with a few retrieved examples (few-shot).

Model	Method	BLEU	ROUGEL-F1	BERTScore	BLEURT	QualityAdapt
Ours (best)	all fine-tuned	**0.1374**	**0.2976**	**0.3359**	**0.5033**	0.9443
GPT-3.5-turbo	zero-shot	0.0288	0.1761	0.1971	0.4638	**0.9765**
	few-shot	0.0695	0.2503	0.3162	0.5009	0.9682
GPT-4	zero-shot	0.0192	0.1537	0.1681	0.4581	0.9764
	few-shot	0.0793	0.2532	0.3246	0.4868	0.9521

Table 8: Evaluation using pretrained large language models from OpenAI (GPT-3.5-turbo and GPT-4) on Taskmaster. We compare using no examples (zero-shot) against prompting with a few retrieved examples (few-shot).

Model	Method	BLEU	ROUGEL-F1	BERTScore	BLEURT	QualityAdapt
Ours (best)	fine-tuned retrieval	0.0441	0.1226	0.2368	0.3710	0.8669
GPT-3.5-turbo	zero-shot	0.0183	0.1260	0.1821	**0.4494**	0.9556
	few-shot	0.0330	0.1641	0.2498	0.4463	0.9360
GPT-4	zero-shot	0.0157	0.1191	0.1637	0.4453	**0.9649**
	few-shot	**0.0444**	**0.1679**	**0.2657**	0.4280	0.9054

formance than the OpenAI models.

Once again, we ran the same evaluation but with the Taskmaster-2 dataset. We compared our approach of using the pretrained FLAN-T5 large model combined with a fine-tuned retrieval component, against GPT-3.5-turbo and GPT-4. In this case, where none of the models has seen data from Taskmaster during training, the OpenAI models achieved better overall performance. These results highlight the generalization capabilities of the GPT models when compared with FLAN-T5 large.

5.6 Delexicalized dataset

Previous studies that work with the MultiWOZ dataset often evaluate results in a delexicalized setting, where named entities are replaced by the corresponding tags according to the span annotations of the dataset (Nekvinda and Dušek, 2022). Although we did not focus on delexicalized datasets, we still tried our proposed system in the delexicalized version of MultiWOZ. Table 11 reports the results for response generation obtained using the standardized MultiWOZ Evaluation script (Nekvinda and Dušek, 2021). Contrary to the results reported in Table 5, introducing retrieved answers in the generation prompt does not increase the obtained BLEU score. We conjecture that this happens because the delexicalized versions of the responses are closer to answer templates and, therefore, simpler than the full responses. The retrieved responses might be only useful to obtain factual information about the named entities, which is unnecessary because the answers are delexicalized.

6 Discussion

6.1 Computational and API costs

We experimented both with models trained and tested in local machines and with models executed online through a paid API from OpenAI. When running our models offline, we consider the computational costs associated with inference and training.

Table 9 shows the total and average times observed. Although these times are highly dependent of the hardware used, we argue they can be compared to better grasp the efficiency of these models. For training, the total time is measured when the training loop is finished due to reaching the maximum number of epochs or the model's performance not improving after some patience

Table 9: Total time elapsed during training and average time per sample during testing on MultiWOZ.

Model	Training	Testing
	total	per sample
all-mpnet-base-v2	3h20	0.006 s
multi-qa-mpnet-base-dot-v1	3h15	0.006 s
FLAN-T5 (small)	2h23	0.09 s
FLAN-T5 (small) w/ retrieval	3h30	0.13 s
FLAN-T5 (large)	2h06	0.94 s
FLAN-T5 (large) w/ retrieval	14h23	0.49 s
FLAN-T5 (XL)	-	0.82 s
FLAN-T5 (XL) w/ retrieval	-	0.80 s
GPT-3.5-turbo	-	4.10 s
GPT-3.5-turbo w/ retrieval	-	2.27 s
GPT-4	-	10.29 s
GPT-4 w/ retrieval	-	5.26 s

steps. Note that when measuring the time for models "with retrieval", we only measure the time of the generation step (with a longer input).

From our measurements, we observe that the retrieval step introduces a very small overhead when compared to generation. Note that this does not include the time necessary to index the databases, which in our experiments took around 1-2 minutes for MultiWOZ. Regarding FLAN-T5, as expected, the larger the version, the longer it takes to run. Smaller batch sizes also contributed to increased times. The sizes of the generated answers also affect the time taken by the models. When introducing retrieval, the generation times tend to decrease because the sizes of the generated answers are shorter (see Table 13 in the appendix).

Regarding GPT-3.5-turbo and GPT-4, we noticed that the times for each call could vary across different runs. Moreover, some of the requests are throttled and we were required to repeat some calls. Compared to our locally run models, all OpenAI's models took much longer to generate the answer.

We also measured the monetary costs associated with calling GPT-3.5 and GPT-4 during the evaluation on MultiWOZ, and report them in Table 10. In our experiments, the cost of GPT-4 was about $15\times$ the cost of GPT-3.5-turbo. Since the performance of GPT-4 is not always better than that of GPT-3.5, the latter model would be a more efficient choice for the considered task-oriented dialogues.

6.2 Dialogue evaluation

During our analysis, we noticed a lack of an all-around satisfactory automatic evaluation metric. The main issue we observed was that the metrics that take a reference and a prediction fail to consider alternative answers that might also be appropriate. Take the following example from MultiWOZ where we used our best model:

Table 10: Cost of OpenAI API (as of May 2023) when evaluating on the MultiWOZ dataset.

Model	Cost	
	per sample	total
GPT-3.5-turbo	$0.0003	$2.41
GPT-3.5-turbo w/ retrieval	$0.0006	$4.15
GPT-4	$0.0051	$37.33
GPT-4 w/ retrieval	$0.0081	$59.94

Context: *User*: Yes, that was all I needed. Thank you very much!
Reference: *System*: Thank you for using our system.
Prediction: *System*: ok, have a great day!

In this simple example, although different, both the reference and prediction are plausible answers. However, lexical-based metrics would have a score of 0 since there are no common words, and even semantic-based metrics would have a hard time because the reference is thanking the user and the prediction is wishing them a great day.

Thus, automatic metrics like QualityAdapt, and similar ones like USR (Mehri and Eskenazi, 2020) and LLM-Eval (Lin and Chen, 2023), can be more appropriate because they do not require a reference and only consider context and prediction. Still, their performance depends greatly on the alignment between training and evaluation data.

6.3 Data leakage

Regarding the GPT-3.5 and GPT-4 models, we do not know what data was used for training. This is especially important when the evaluation is performed with public datasets, since these models might have already seen this data. In the case of FLAN-T5, the authors report using the Taskmaster dataset for training. Most likely, only the train split was used. Nonetheless, the GPT models might have an unfair advantage over FLAN-T5.

7 Conclusions

We performed a systematic evaluation of different ways of using state-of-the-art retrieval and generation models for task-oriented answer generation. We experimented with dense retrieval models, FLAN-T5, GPT-3.5, and GPT-4, evaluating them on the MultiWOZ and Taskmaster-2 datasets. Having explored these models separately and combined, we concluded that retrieving possible answers greatly improved the generated responses in terms of automatic metrics. Moreover, if training data is available and it does not differ much from

the data seen during inference, then fine-tuning the generation model can greatly improve its performance, surpassing strong results from large language models such as GPT-3.5 and GPT-4. If the dialogue system is to be used in a context of high variability, then using a more general large language model and only fine-tuning the retrieval component can be a better procedure.

In future work, we shall test the generation model with other prompts and evaluate how the performance is affected. Moreover, we plan to improve the training of the retrieval model, since it can be integrated with any generation system and, as we have shown, significantly improve its performance. Training with different datasets can improve its generalization ability, and strategies like maximal marginal relevance (Carbonell and Goldstein, 1998) can be used to collect diverse answers. Lastly, active retrieval augmentation methods similar to FLARE (Jiang et al., 2023) can also be employed. This involves generating initial answers from the context (without retrieval), refining the retrieval query with these generated answers, and, lastly, re-generating the final answer with the retrieved candidates.

Acknowledgements

We would like to express our sincere gratitude to Jamie Callan for hosting Gonçalo Raposo at LTI – CMU during a three-month summer internship and for his valuable collaboration in the research presented in this paper. His guidance and support have been instrumental in the success of this work.

This research was supported by the Portuguese Recovery and Resilience Plan through the project C645008882-00000055 (Center for Responsible AI), and through *Fundação para a Ciência e a Tecnologia* (FCT), specifically through the P2020 program LISBOA-01-0247-FEDER-045909 (MAIA), and through the INESC-ID multi-annual funding with reference UIDB/50021/2020.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Ben Goodrich, Daniel Duckworth, Semih Yavuz, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4516–4525, Hong Kong, China. Association for Computational Linguistics.

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 335–336, New York, NY, USA. Association for Computing Machinery.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *preprint*. arXiv:2210.11416.

Jianfeng Gao, Chenyan Xiong, Paul Bennett, and Nick Craswell. 2023. *Neural Approaches to Conversational Information Retrieval*. Springer International Publishing.

Prakhar Gupta, Cathy Jiao, Yi-Ting Yeh, Shikib Mehri, Maxine Eskenazi, and Jeffrey Bigham. 2022. InstructDial: Improving zero and few-shot generalization in dialogue through instruction tuning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 505–525, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot Learning with Retrieval Augmented Language Models. *preprint*. arXiv:2208.03299.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. *preprint*. arXiv:2305.06983.

Jeff Johnson, Matthijs Douze, and Herve Jegou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

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

Yen-Ting Lin and Yun-Nung Chen. 2023. Llm-eval: Unified multi-dimensional automatic evaluation for open-domain conversations with large language models. *preprint*. arXiv:2305.13711.

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. 2020. MinTL: Minimalist transfer learning for task-oriented dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Shikib Mehri and Maxine Eskenazi. 2020. USR: An unsupervised and reference free evaluation metric for dialog generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

John Mendonca, Alon Lavie, and Isabel Trancoso. 2022. QualityAdapt: an automatic dialogue quality estimation framework. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 83–90, Edinburgh, UK. Association for Computational Linguistics.

Tomáš Nekvinda and Ondřej Dušek. 2021. Shades of BLEU, flavours of success: The case of MultiWOZ. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 34–46, Online. Association for Computational Linguistics.

Tomáš Nekvinda and Ondřej Dušek. 2022. AARGH! end-to-end retrieval-generation for task-oriented dialog. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 283–297, Edinburgh, UK. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 Technical Report. *preprint*. arXiv:2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Baolin Peng, Michel Galley, Pengcheng He, Chris Brockett, Lars Liden, Elnaz Nouri, Zhou Yu, Bill Dolan, and Jianfeng Gao. 2022. Godel: Large-scale pre-training for goal-directed dialog. *preprint*. arXiv:2206.11309.

Gustavo Penha and Claudia Hauff. 2023. Do the findings of document and passage retrieval generalize to the retrieval of responses for dialogues? In *Lecture Notes in Computer Science*, pages 132–147. Springer Nature Switzerland.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. *preprint*. `https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf`.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *preprint*. arXiv:2302.00083.

Gonçalo Raposo, Rui Ribeiro, Bruno Martins, and Luísa Coheur. 2022. Question rewriting? assessing its importance for conversational question answering. In *Advances in Information Retrieval*, pages 199–206, Cham. Springer International Publishing.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, Morteza Behrooz, William Ngan, Spencer Poff, Naman Goyal, Arthur Szlam, Y-Lan Boureau, Melanie Kambadur, and Jason Weston. 2022. BlenderBot 3: a deployed conversational agent that continually learns to responsibly engage. *preprint*. arXiv:2208.03188.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867. Curran Associates, Inc.

Haipeng Sun, Junwei Bao, Youzheng Wu, and Xiaodong He. 2022. Mars: Semantic-aware contrastive learning for end-to-end task-oriented dialog. *preprint*. arXiv:2210.08917.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. 2022. LaMDA: Language Models for Dialog Applications. *preprint*. arXiv:2201.08239.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Vaswani, ashish and shazeer, noam and parmar, niki and uszkoreit, jakob and jones, llion and gomez, aidan n and kaiser, łukasz and polosukhin, illia. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Qi Wang, Yue Ma, Kun Zhao, and Yingjie Tian. 2020. A comprehensive survey of loss functions in machine learning. *Annals of Data Science*, 9(2):187–212.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Liu Yang, Junjie Hu, Minghui Qiu, Chen Qu, Jianfeng Gao, W. Bruce Croft, Xiaodong Liu, Yelong Shen, and Jingjing Liu. 2019. A hybrid retrieval-generation neural conversation model. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. ACM.

Xiaoxue Zang, Abhinav Rastogi, and Jindong Chen. 2020. MultiWOZ 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open pre-trained transformer language models. *preprint*. arXiv:2205.01068.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Zheng Zhang, Ryuichi Takanobu, Qi Zhu, MinLie Huang, and XiaoYan Zhu. 2020. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, 63(10):2011–2027.

A Additional Tables

Table 11: Results on the delexicalized version of MultiWOZ. We report the top end-to-end generation model from the MultiWOZ benchmark and our best fine-tuned versions of a retrieval-only system, a generation-only system, and a system combining both retrieval and generation. Our generation-only system obtained a score similar to the top model from the benchmark, which can be expected given the similarities of these approaches.

Method	**BLEU**
Mars (Sun et al., 2022)	0.199
Retrieval-only	0.1091
Generation-only	0.1969
Retrieval + Generation	0.1790

Table 12: Average and total times measured during the training and testing of the evaluated models on the MultiWOZ dataset. Our local models were executed using an `NVIDIA Quadro RTX 6000` GPU with 24,GB of memory. Variations in the measured times can be attributed to differences in model sizes, batch sizes, input and output sizes, among other factors. Additionally, the times of the OpenAI models exhibited variability across different runs, possibly resulting from high demand.

Model	**Training**		**Testing**	
	per sample	**total**	**per sample**	**total**
all-mpnet-base-v2	0.01 s	3h20	0.006 s	41 s
multi-qa-mpnet-base-dot-v1	0.01 s	3h15	0.006 s	44 s
FLAN-T5 (small)	0.06 s	2h23	0.09 s	10 m
FLAN-T5 (small) w/ retrieval	0.17 s	3h30	0.13 s	15 m
FLAN-T5 (large)	1.81 s	2h06	0.94 s	1h56
FLAN-T5 (large) w/ retrieval	5.27 s	14h23	0.49 s	1h01
FLAN-T5 (XL)	-	-	0.82 s	1h41
FLAN-T5 (XL) w/ retrieval	-	-	0.80 s	1h38
GPT-3.5-turbo	-	-	4.10 s	8.39 h
GPT-3.5-turbo w/ retrieval	-	-	2.27 s	4.65 h
GPT-4	-	-	10.29 s	21.08 h
GPT-4 w/ retrieval	-	-	5.26 s	10.77 h

Table 13: Number of input and generated tokens obtained from the OpenAI models, along with the associated cost of their API usage (as of May 2023) during the evaluation on the MultiWOZ dataset. The cost calculation follows the information provided in the OpenAI documentation, which involves tokenizing the input and output texts, counting the number of tokens, and multiplying by the corresponding cost per token.

Model	**Input tokens**		**Generated tokens**		**Cost**	
	Average	**Total**	**Average**	**Total**	**Average**	**Total**
GPT-3.5-turbo	109.6	808K	53.8	397K	$0.0003	$2.41
GPT-3.5-turbo w/ retrieval	243.6	1 796K	37.6	277K	$0.0006	$4.15
GPT-4	109.6	808K	53.8	397K	$0.0051	$37.33
GPT-4 w/ retrieval	243.6	1 796K	37.6	277K	$0.0081	$59.94

B Examples

Table 14: Two random examples from the evaluation on the MultiWOZ dataset. Each example includes the conversation context and ground-truth response, followed by the answers obtained using the methods covered in this work. The reported retrieval approach utilizes the model that indexes the answers. These examples illustrate the impact of context size on each method and highlight the verbosity of OpenAI models.

Sample	**pmul4819_0-2**	**mul1285_2-8**
Context	User: I'm needing a train leaving on wednesday and arriving by 11:00.	User: I would like something in the cheap range and i need the address and phone number System: Sure, how about Pizza Hut City Centre? It's at Regent Street City Centre cb21ab and can be reached at 01223323737. May I help with anything else? User: Yes, please. I would like information about the Ashley Hotel. System: The Ashley Hotel is in the north part of town. It is a moderately priced two star hotel with internet service and parking. Would you like to make a reservation? User: Yes, 3 people, 5 nights starting from wednesday. Reference number please.
Response	System: Sure, where are you leaving from and traveling to?	System: Booking was successful. Reference number is KJGGRH7E .
Retrieval	System: There is a train on Wednesday at 11:01 am.	System: Ashley Hotel is in the moderate price range. Their address is 74 Chesterton Road and the number is 01223350059. Anything else I can assist you with, today?
Fine-tuned retrieval	System: I have 318 trains running on Wednesday, to help narrow it down where will you be departing/arriving?	System: The Ashley Hotel's book was unsuccessful. Would you like me to try Lovell Lodge?
FLAN-T5 (large)	System: i can help you with that. What time would you like to leave and where would you like to arrive?	System: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa (...)
Fine-tuned FLAN-T5 (large)	System: Certainly. Where will you be departing from?	System: Certainly. Your booking was successful. Your Reference number is : YYYYYYYC. Is there anything else I can help you with?
FLAN-T5 (large) w/ retrieval	System: ok. I can help you narrow your search if you give me your departure site and your destination?	System: X21XYR7K
Fine-tuned FLAN-T5 (large) w/ retrieval	System: Certainly. Where will you be departing from?	System: i was able to book your party of 3 for 5 nights starting Wednesday at the Ashley Hotel. your reference number is : YYYYZZ1Y.
GPT-3.5-turbo	Customer service: Hello! Thanks for reaching out to us. May I know your starting location and destination? This will help me give you the right information.	System: I'm sorry, as a customer service system, I do not have the capability to make reservations. However, I can suggest that you visit the Ashley Hotel's website or give them a call at +44 (0) 1223 293383 to make your booking. They can also provide you with a reference number once your reservation is confirmed. Is there anything else I can assist you with?
GPT-3.5-turbo w/ retrieval	System: There is a train leaving from London Kings Cross at 11:00 and arrives by 11:51. Would you like me to check availability for you?	System: No problem. I have made a booking for 3 people at the Ashley Hotel for 5 nights starting from Wednesday. Your reference number is X21XYR7K. Is there anything else I can assist you with?
GPT-4	Hello! I'd be happy to help you find a train that meets your requirements. Can you please provide me with the departure and destination stations so I can look up train schedules for you?	System: Great! I've made a reservation for 3 people for 5 nights starting from Wednesday at the Ashley Hotel. Your reference number is AHSY12345. Please keep this number for future reference. If you have any questions or need to make changes, feel free to reach out. Enjoy your stay!
GPT-4 w/ retrieval	System: There are 404 trains running on Wednesday, what point of departure and point of arrival are you interested in?	System: I have booked it for 5 nights/8 people starting Wednesday. The booking reference is KX7LBOB7 . Anything else?