

Are Large Language Models All You Need for Task-Oriented Dialogue?

Vojtěch Hudeček and Ondřej Dušek

Charles University, Faculty of Mathematics and Physics
Malostranské náměstí 25, 118 00 Prague, Czechia
hudecek@ufal.mff.cuni.cz, odusek@ufal.mff.cuni.cz

Abstract

Instruction-finetuned large language models (LLMs) gained a huge popularity recently, thanks to their ability to interact with users through conversation. In this work, we aim to evaluate their ability to complete multi-turn tasks and interact with external databases in the context of established task-oriented dialogue benchmarks. We show that in explicit belief state tracking, LLMs underperform compared to specialized task-specific models. Nevertheless, they show some ability to guide the dialogue to a successful ending through their generated responses if they are provided with correct slot values. Furthermore, this ability improves with few-shot in-domain examples.

1 Introduction

Large Language Models (LLMs) have transformed the NLP field, showing outstanding performance across many NLP benchmarks such as Winograd Challenge (Levesque et al., 2012) or GLUE (Wang et al., 2018). Recently, instruction finetuning of LLMs proved to be able to align the model outputs with human preferences (Ouyang et al., 2022; Wang et al., 2022) and improved the LLMs’ communication capabilities substantially. State-of-the-art LLMs are not only good at understanding user needs but also capable of providing relevant answers. Consequently, we see many chatbot applications both inside and outside academia (ChatGPT¹, Claude², Sparrow³) which build upon the raw power of instruction-finetuned LLMs.

Given the millions of daily interactions with these chatbots, it appears that the models are able to handle users’ needs to their satisfaction, at least to some extent. However, these chatbots are tuned using unstructured open-domain conversations. The

¹<https://openai.com/blog/chatgpt>

²<https://www.anthropic.com/index/introducing-claude>

³<https://www.deepmind.com/blog/building-safer-dialogue-agents>

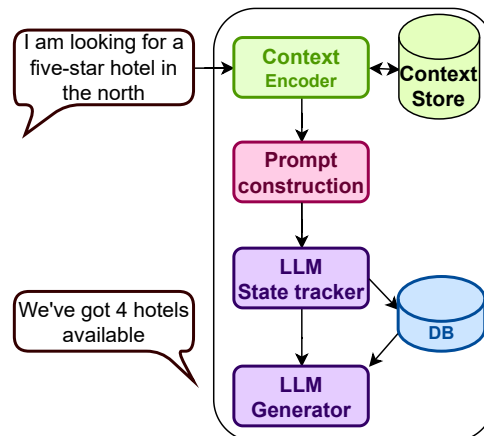


Figure 1: A high-level overview of our proposed pipeline. The user input is used to retrieve relevant few-shot examples (if available). Then, an initial prompt is constructed and an LLM is asked to provide the current dialogue state. Based on that, we retrieve database results and construct another prompt. Finally, we ask the LLM to provide the response.

aim of this paper is to evaluate these systems for more specific applications, where the system has to follow a predetermined structure and handle external sources of information, such as APIs or databases. We raise the question to what extent LLMs are capable of handling these applications off-the-shelf, i.e. without finetuning. We thus choose to evaluate LLM performance in the task-oriented dialogue (TOD) setting, as it requires precise information handling for communicating with external APIs. Moreover, TOD systems output in-domain information which has predetermined structure and lends itself well to evaluation, thanks to pre-existing annotated data sets. We avoid any finetuning techniques and focus on zero-shot or few-shot settings using in-context learning, as this approach has lower hardware requirements and barrier of entry and better flexibility or even performance in certain tasks (Su et al., 2022).

Therefore, we introduce an LLM-based TOD

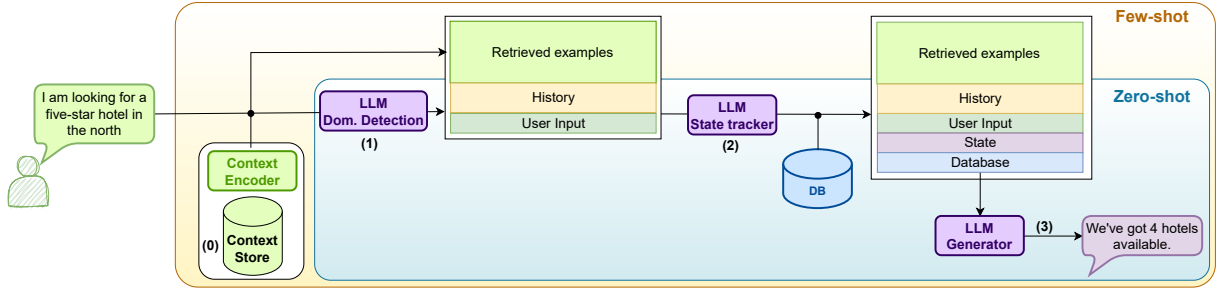


Figure 2: A detailed description of our proposed pipeline. (0) As a preprocessing step, we encode a subset of the training set that will be used to retrieve few-shot examples. Given the user input, we: (1) Detect the domain, retrieve relevant examples (in the few-shot setting) and construct an initial prompt. (2) Infer the belief state using LLM. Based on that, we retrieve database information and construct another prompt that includes both the state and database results. (3) We ask the LLM to provide a final response.

conversation pipeline (see Figure 1) and evaluate its performance with respect to commonly used task-oriented metrics such as Joint Goal Accuracy, Slot F1, and Dialogue Success (Rastogi et al., 2018; Budzianowski et al., 2018). Our pipeline resembles other approaches based on LMs (Peng et al., 2021; Yang et al., 2021), using state tracking and response generation as two main, separate steps, while keeping the role of a dialogue policy implicit. However, instead of finetuning LMs, it intentionally relies almost exclusively on the usage of pretrained LLMs as-is, so we can test their out-of-the-box capabilities. The dialogue context and domain description are introduced to the model only by including them in the input prompt. In the zero-shot setting, the model receives a domain description only; in the few-shot setting, it additionally uses a few retrieved examples (see Section 3 for details).

In our experiments, we find that LLMs are not very good at state tracking and their performance falls behind the state-of-the-art. However, if provided with correct belief states, some of them yield interesting response generation performance, comparable to earlier finetuned state-of-the-art models. To our knowledge, our zero-shot experiments establish a state-of-the-art result in unsupervised TOD modeling on the MultiWOZ and Schema-guided datasets (Budzianowski et al., 2018; Rastogi et al., 2020). While there may be room for improvement through prompt engineering, our results aim to show the out-of-the-box LLM capabilities. We plan to release our experimental code on GitHub.⁴

⁴<https://github.com/vojtsek/to-llm-bot>

2 Related Work

Large Language Models The Transformer architecture (Vaswani et al., 2017) enabled the training of large and capable language models. The research on their few-shot and zero-shot abilities dates back to the GPT-2 and GPT-3 models (Radford et al., 2019; Brown et al., 2020), which are scaled versions of the Transformer decoder. Many followed this path of training large Transformer decoders (Zhang et al., 2022; Black et al., 2022), yielding models of up to hundreds of billions parameters in size (Zhao et al., 2023). Other models leverage the whole original (encoder-decoder) Transformer architecture (Raffel et al., 2020; Lewis et al., 2020). Recent research focuses on improving the training of moderate-sized architectures to broaden access to highly capable LLMs (Touvron et al., 2023).

Instruction Tuning The idea of using reinforcement learning techniques to align model-based agents better with users’ intents was pioneered in game agent development (Christiano et al., 2017) and later explored for training language models (Ziegler et al., 2019; Ouyang et al., 2022). Although these techniques proved to be quite effective, the process is still very demanding in terms of collecting feedback from users. Consequently, several datasets were proposed (Wang et al., 2022; Iyer et al., 2022; Black et al., 2022) that collected millions of instructions-based tasks in natural language and can be applied to finetune LLMs using reinforcement learning.

LM-based TOD modeling Task-oriented dialogue modeling with pretrained LMs was introduced by Zhang et al. (2019) and Peng et al. (2021), who followed text-based state encoding and two-

stage generation proposed by [Lei et al. \(2018\)](#): An LM is first used to decode a structured belief state, represented as text. The belief state is then used to retrieve database information and the LM is called once more to generate a response, conditioned on the belief state and retrieved information. Several improvements to the basic setup were proposed, such as contrastive state training ([Kulhánek et al., 2021](#)) or using belief state differences ([Lin et al., 2020](#)). Others proposed a combination of generative models with retrieval-based approaches ([Pandey et al., 2018](#); [Cai et al., 2019](#); [Nekvinda and Dušek, 2022](#)). All described works finetune LMs on in-domain data, which is in contrast with the pure in-context learning approach that we apply.

Few-shot dialogue modeling One of the first neural models focusing on learning dialogue from a few in-domain examples was the Hybrid Code Networks ([Williams et al., 2017](#)), a trainable system based on recurrent neural networks, with partially handcrafted components. Another approach was proposed by [Zhao and Eskenazi \(2018\)](#), who used latent action representations to enable the transfer of domain knowledge. Latent actions were also used by [Huang et al. \(2020\)](#) and [Shalyminov et al. \(2019\)](#). More recent approaches leverage the capabilities of pretrained Transformer LMs ([Shalyminov et al., 2020](#)). [Hu et al. \(2022\)](#) used LLMs and in-context learning to perform belief state tracking, formulating the task as an SQL query generation. Unlike our work, they did not use instruction-tuned models and omitted database retrieval and response generation.

3 Method

We introduce our method step-by-step. An overall description of the proposed pipeline is shown in [Figure 2](#). The system consists of a pretrained LLM and an (optional) context store in a vector database. Three LLM calls are performed in each dialogue turn, with specific prompts (see [Section 3.1](#)). First, the LLM performs domain detection and state tracking ([Section 3.2](#)). The updated belief state informs a database query, whose results are used in the subsequent LLM-based response generation step ([Section 3.3](#)). In the few-shot setting, the context store is used to store a limited number of examples from the training set, which are retrieved based on similarity with the conversation context and included in LLM prompts (see [Section 3.4](#)).

Prompt	Definition: Capture values from a conversation about hotels. Capture pair "entity:value" separated by colon and no spaces in between. Separate the "entity:value" pairs by hyphens Values that should be captured are: - "pricerange": the price of the hotel ... [history] Customer: "I want a cheap place to stay."
Output:	pricerange:"cheap"

Table 1: A simplified example of a zero-shot version of the prompt used for state update prediction. It contains **task definition**, **domain description**, **dialogue history** and **user utterance**. For the exact prompts see [Appendix](#).

3.1 Prompt construction

We aim to compare the raw capabilities of the selected LLMs, therefore we do not focus on prompt engineering techniques and choose universal prompts used for all LLMs in this work (cf. [Section 8](#)). We choose simple, plain language statements as prompts, with no specific vocabulary, based only on a few preliminary tests. We define a single **domain detection prompt** for all examples, plus a pair of prompts for each domain in the given dataset: a **state tracking prompt** (see [Table 1](#)) and a **response prompt**.

The domain detection prompt includes a task description and two static examples of domain detection. In addition to general instructions, each state tracking prompt contains a domain description, a list of relevant slots, the dialogue history, and the current user utterance. The response prompts do not contain the per-domain slot list, but they include the current belief state and database results instead. In the few-shot setting, each tracking and response prompt additionally contains positive and negative examples retrieved from the context store (see [Section 3.4](#)). Prompt examples are shown in [Tables 5](#) and [6](#) in the [Appendix](#).

3.2 Domain Detection and State Tracking

We prompt the LM twice at each turn during state tracking: first, to detect the active domain, then to output slot values that changed or appeared in the current turn. We then use the outputs to update the accumulated global belief state.

The two prompting steps are used since we need the models to operate in a multi-domain setting, i.e., handle conversations spanning multiple domains. Therefore, we need to be able to detect the currently

active domain. We achieve this by first prompting the LLM with a domain detection prompt (using a single prompt for all examples).

Once we obtain the active domain prediction, we can include manually designed domain descriptions in a second prompt that handles belief state prediction. An example of a prompt used for state tracking is provided in Table 1. For the few-shot variants, we retrieve few-shot examples from the context store, limited to the active domain.⁵

Our preliminary experiments showed that LLMs struggle to output all active slot values at every turn consistently. Therefore, we model only state updates, following the MinTL approach (Lin et al., 2020). Here, the model only generates the slot-value pairs that have changed in current turn. The global belief state is then accumulated using these turn-level updates. To obtain machine-readable outputs useful for database queries or API calls, we specify in the prompt that the model should provide JSON outputs, and any provided few-shot examples are formatted accordingly.

3.3 Response Generation

The current belief state is used to query the database for entries matching all user-specified slots in the active domain. Given the belief state and database results, the response generation is straightforward. The prompt for the LLM includes dialogue history, user utterance, belief state and database results (and retrieved examples in the few-shot setting) and requests the model to provide a fitting system response. We generate delexicalized responses (Wen et al., 2015), i.e., we replace slot values by placeholders, following prior work in end-to-end TOD modeling. In addition to simplifying the task for the model, delexicalized outputs allow us to evaluate the success rate and compare to previous works. The prompt specifies that the model should provide entity values as delexicalized placeholders, and any few-shot examples are constructed accordingly.

3.4 Context Storage

It has been shown that enriching prompts with specific examples boosts LM performance (Madotto et al., 2020; Brown et al., 2020). To apply this knowledge efficiently in our pipeline, we introduce a storage that contains encoded dialogue contexts.

⁵For this purpose, each conversation snippet contained in the context store comes from a single-domain conversation.

This context storage is optional and is only required for the few-shot prompting variant. We use dialogue context taken from a fixed-length history window as the key to be encoded in the vector database. More details can be found in Section 4.4. Once the relevant examples are retrieved, we include them in the prompt to guide the model better. Some of the LLMs rely on negative (counter-) examples as well (Wang et al., 2022). Therefore, we follow Peng et al. (2021)’s consistency classification task approach to produce negative examples: We take some of the retrieved belief state examples, corrupt them by replacing some of the correct slot values with random values, and present them as negative in the prompt.

4 Experimental Setup

To obtain a broad overview of the current LLMs’ capabilities, we compare several models, spanning different numbers of trainable parameters and different training methods. We also experiment with four variants of the base setup, using either zero-shot or few-shot operations and using either predicted or oracle belief states.

4.1 Datasets

We experiment with two of the currently most prominent benchmark datasets for task-oriented multi-domain dialogue:

- **MultiWOZ 2.2** (Budzianowski et al., 2018; Hung et al., 2022) is a well-known benchmark used for evaluating state tracking, response generation and dialogue success rate. Its evaluation is well-defined and the dataset contains database files, so full interaction can be simulated. It contains over 10k dialogues, 7 domains and 29 distinct slots.
- **Schema Guided Dataset** (Rastogi et al., 2020) is also well annotated and even richer dataset containing more than 22k dialogues 18 domains and 145 slots. Database interaction is considered in the dataset, but no real database is provided and database results are defined ad-hoc. Therefore we simply use the provided database results in the prompts without performing any actual queries.

4.2 Tested Models

We chose the following five instruction-finetuned models for our experiments, spanning different

model	few shot	oracle BS	Schema Guided Dialogues				MultiWOZ 2.2			
			BLEU	JGA	Slot-F1	Success	BLEU	JGA	Slot-F1	Success
Supervised SotA	✗	✗	29.90*	0.30 [†]	0.60*	–	19.90 [♣]	0.60 [◇]	–	0.82 [♡]
<i>Alpaca-LoRA-7B-zs-gbs</i>	✗	✗	2.79	0.02	0.01	0.11	1.61	0.06	0.07	0.04
<i>Tk-Instruct-11B-zs-gbs</i>	✗	✗	4.16	0.05	0.03	0.10	2.48	0.04	0.04	0.04
<i>GPT-NeoXT-20B-zs-gbs</i>	✗	✗	0.45	0.01	0.01	0.17	0.52	0.03	0.02	0.04
<i>OPT-IML-30B-zs-gbs</i>	✗	✗	1.63	0.01	0.01	0.17	0.56	0.02	0.04	0.03
<i>ChatGPT-zs-gbs</i>	✗	✗	–	–	–	–	4.17	0.13	0.40	0.31
<i>Alpaca-LoRA-7B-zs-obs</i>	✗	✓	2.76	–	–	0.23	1.73	–	–	0.08
<i>Tk-Instruct-11B-zs-obs</i>	✗	✓	5.21	–	–	0.24	2.66	–	–	0.18
<i>GPT-NeoXT-20B-zs-obs</i>	✗	✓	0.83	–	–	0.22	0.60	–	–	0.06
<i>OPT-IML-30B-zs-obs</i>	✗	✓	1.94	–	–	0.22	0.54	–	–	0.06
<i>ChatGPT-zs-obs</i>	✗	✓	–	–	–	–	3.76	–	–	0.47
<i>Alpaca-LoRA-7B-fs-gbs</i>	✓	✗	6.32	0.04	0.01	0.09	5.53	0.06	0.08	0.06
<i>Tk-Instruct-11B-fs-gbs</i>	✓	✗	6.66	0.06	0.05	0.10	6.56	0.16	0.33	0.19
<i>GPT-NeoXT-20B-fs-gbs</i>	✓	✗	1.62	0.04	0.02	0.09	2.73	0.05	0.04	0.05
<i>OPT-IML-30B-fs-gbs</i>	✓	✗	0.82	0.06	0.07	0.08	4.40	0.03	0.03	0.04
<i>ChatGPT-fs-gbs</i>	✓	✗	–	–	–	–	6.77	0.27	0.51	0.44
<i>Alpaca-LoRA-7B-fs-obs</i>	✓	✓	6.99	–	–	0.25	5.96	–	–	0.41
<i>Tk-Instruct-11B-fs-obs</i>	✓	✓	8.56	–	–	0.25	6.91	–	–	0.46
<i>GPT-NeoXT-20B-fs-obs</i>	✓	✓	1.97	–	–	0.24	2.92	–	–	0.28
<i>OPT-IML-30B-fs-obs</i>	✓	✓	0.56	–	–	0.22	5.40	–	–	0.28
<i>ChatGPT-fs-obs</i>	✓	✓	–	–	–	–	6.84	–	–	0.68

Table 2: Evaluation of the chosen LLMs with respect to widely used TOD measures. For each model, we provide multiple variants. We use either zero-shot or few-shot prompts (-zs- vs. -fs-) and either generated or oracle belief state (-gbs vs. -obs). The few-shot variants use 10 examples per domain in the context storage (~0.6% of the training set in case of MultiWOZ), two of which are selected for the prompts. To reduce cost, we only evaluate the paid ChatGPT model on MultiWOZ. We also provide supervised state-of-the-art results to put the numbers in context: *Zhu et al. (2022), [†]Feng et al. (2021), [♣]Sun et al. (2022), [◇]Huang et al. (2023), [♡]Feng et al. (2023).

sizes (within the limitations of hardware available to us) and using freely available models as well as the paid ChatGPT API. We indicate the specific model variant (i.e., model size, given by the number of parameters) directly in the model name.

- **Tk-Instruct-11B** (Wang et al., 2022) is based on the T5 encoder-decoder architecture (Rafael et al., 2020). It was tuned on a dataset of over 5M task instances with instructions.
- **ChatGPT** is a product introduced by OpenAI.⁶ Although the exact training process and architectures were not published, it most probably uses a similar architecture and finetuning techniques as InstructGPT (Ouyang et al., 2022), with additional human feedback.
- **Alpaca-LoRA-7B** is a version of the LLaMa model (Touvron et al., 2023) using the LoRA method (Hu et al., 2021) for finetuning on Stanford Alpaca project data (Taori et al., 2023). LoRa keeps the base model parameters frozen, but adds additional smaller weight matrices to the model to transform its outputs.

⁶<https://openai.com/blog/chatgpt>

- **GPT-NeoXT-Chat-Base-20B** is based on the GPT-NeoX open-source language model (Black et al., 2022) and finetuned with over 40M dialogue-style instructions.
- **OPT-IML-30B** (Iyer et al., 2022) is based on the Transformer decoder OPT model (Zhang et al., 2022) and trained with a custom set of instructions, including the finetuning set from Tk-Instruct.

4.3 Evaluated variants

We test four variants of our setup for each pair of model and dataset. Specifically, we use zero-shot (without examples) or few-shot (including examples) prompts (-zs- vs. -fs-) and either generated or oracle belief states (-gbs vs. -obs). For retrieval in the few-shot setting, we store just 10 examples per domain in the context store by default. We experiment with increasing this number in Section 5.4. Using oracle belief state allows us to focus on evaluating the LLM’s ability to guide the dialogue.

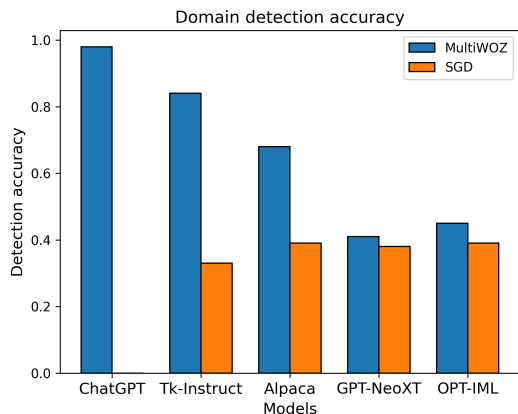


Figure 3: Domain detection accuracy with respect to different models for MultiWOZ 2.2 and SGD data which consist of 7 and 18 domains, respectively.

4.4 Experiment Details

Due to the expensiveness of the LLM runs,⁷ we did not perform a grid search, but used a limited set of preliminary experiments to determine hyperparameters. Based on this, we used the context of two preceding utterances (user + system) as the context store keys (cf. Section 3.4). We retrieve two examples for few-shot prompts and make one corrupted variant from each of them for negative examples. To corrupt an example, we switch some of the slot values randomly, similarly to Kulhánek et al. (2021). In the context store, we encode few-shot examples using the multilingual embedding model provided by Reimers and Gurevych (2020)⁸ and store them in the FAISS database (Johnson et al., 2019). To perform the LLM calls, we use the Huggingface library⁹ and the OpenAI API.¹⁰

4.5 Evaluation Measures

We evaluate the system outputs on multiple levels, both using automatic metrics and human evaluation. Results are given in Sections 5 and 6, respectively.

Automatic Metrics

In automatic evaluation, we first follow the LLM calls being made and evaluate domain detection, state tracking as well as response generation. We also evaluate the overall dialogue-level performance. For *domain detection*, we simply compute

⁷Hardware intensity for the freely available models and actual cost for ChatGPT.

⁸<https://huggingface.co/sentence-transformers/all-mpnet-base-v2>

⁹<https://huggingface.co>

¹⁰<https://platform.openai.com>

detection accuracy as a ratio of correctly detected domain out of all dialogue turns being processed. For *state tracking*, we compute **micro-F1** score and **Joint Goal Accuracy (JGA)**. JGA is computed as the ratio of dialogue turns for which the predicted belief state matches the ground truth. We use fuzzy matching of the slot values, so that capitalization or minor typos do not influence the result. To evaluate *response generation*, we follow related works and use **BLEU score** (Papineni et al., 2002).

The main *overall measure* for evaluating a task-oriented dialogue is the dialogue **success rate** (De-riou et al., 2021). For MultiWOZ, we use the standard evaluation of dialogue success as the ratio of dialogues where the user reaches the desired goal, based on goal annotation provided with the data (Nekvinda and Dušek, 2021). The SGD dataset does not include goal annotation but contains information about the requested slots. Therefore, we compute SGD success rate as the proportion of dialogues in which (1) the system captures all the slots correctly and (2) all the requested slots are provided.

Human Evaluation

For human evaluation, we perform a small-scale in-house interaction study on MultiWOZ. Since the MultiWOZ goal often involves tasks in multiple domains, we ask annotators to evaluate each domain in the dialogue distinctly. At the end of each dialogue, the annotators are asked to answer these questions:

1. *How many of the subdialogues/domains were handled successfully?* (corresponding to dialogue success)
2. *How many clarifications or corrections were needed?*
3. *Was all the provided information captured correctly?* (corresponding to JGA)

5 Automatic Metrics Results

5.1 Domain detection

We report the domain detection accuracy on MultiWOZ and SGD in Figure 3. We observe that the domain detection accuracy varies quite a lot for various models and presumably influences the quality of the retrieved few-shot examples and appropriateness of the subsequent prompts. However, it is important to note that domain detection is turn-based,

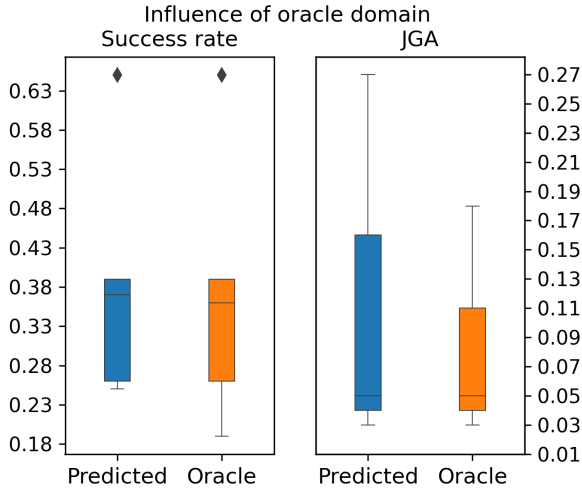


Figure 4: The influence of using oracle domain to retrieve examples. Interestingly, the oracle domain does not improve the performance, suggesting that the model-based detection is good enough for retrieval.

and arguably there are situations (e.g. providing an address, saying goodbye etc.) that are always handled in the same fashion, even though they formally belong to different domains. Therefore, not all the retrieved examples from misclassified domains necessarily contain unrelated contexts. To explore this, we measure the performance of all models in case an oracle domain is given to them (Figure 4). Interestingly, using the oracle domain did not improve performance, it even worsened in some cases. This suggests that the model-predicted domain is generally good enough, and additionally providing the domain information does not contribute to the final system performance. The negative influence on performance might be caused by forcing the system to filter out relevant examples. We observe that in multiple cases, the conversations snippets are domain-independent so the retrieval might perform better even with a wrongly selected domain. Forcing the ground truth domain examples in these cases can be potentially harmful.

5.2 Belief State Tracking

The belief state tracking results overview is given in Table 2 (*JGA* and *Slot-F1*). There is a huge gap between the supervised models’ performance and the LLM results. Also compared to Hu et al. (2022), who used few-shot in-context learning and reported JGA 43.13% with a comparable dataset size, our instruction-tuned LLMs fall short. However, the models we use are an order of magnitude

Number of stored examples vs. the performance of the model

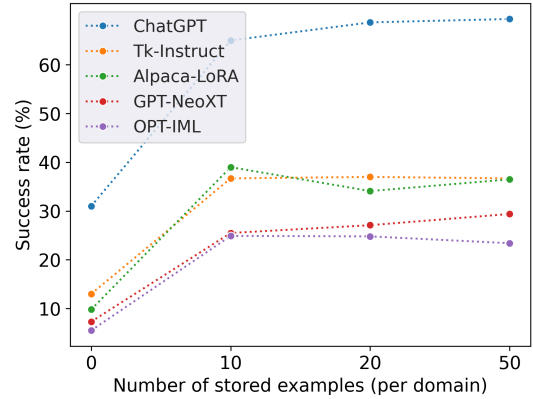


Figure 5: The influence of the number of examples per domain available for few-shot retrieval and performance of the model in terms of the dialogue success on MultiWOZ 2.2 data with oracle state supplied. Note that this does not represent the number of examples selected for the prompt, which is fixed to two.

smaller in general, and we also use fewer examples in the prompt. We hypothesize that the performance could be further improved by careful model-specific prompt customization and perhaps task re-formulation; nevertheless, this is not the goal of this work. We intentionally focus on the universal framing of the task since we want to explore the general ability of the models to follow instructions.

When comparing the results among the models, ChatGPT clearly outperforms the rest of the models by a large margin. Interestingly, the few-shot vs. zero-shot setting does not seem to influence the results much, except for the GPT-NeoXT model.

5.3 Response Generation

BLEU scores are low overall, far below the supervised state-of-the-art. Tk-Instruct and ChatGPT are the strongest here and perform roughly on par.

5.4 Dialogue-level performance

Results for dialogue success are provided in Table 2, and there is again a large gap between LLMs and supervised custom models’ performance. ChatGPT seems to outperform other models, similarly to state tracking (cf. Section 5.2). However, for some cases, especially in the zero-shot setting, the difference is not that obvious. In most cases, adding the retrieved few-shot examples helps. The contribution of retrieved examples is more obvious when we supply the oracle belief state, in which case it helps consistently for all the models.

We also explore the influence of context storage size on the dialogue success rate. The results are given in Figure 5. It seems that the biggest improvement can be achieved by supplying just a few examples instead of zero-shot prompting, but increasing the size of the example pool for retrieval does not yield further performance gains.

6 Model Analysis

6.1 Human Evaluation

We employed 6 annotators with a background in linguistics and NLP and let them interact with the two strongest models in terms of automatic metrics: ChatGPT and Tk-Instruct. The annotators were given randomly selected goals from the MultiWOZ 2.2 dataset and a minimal set of essential instructions on how to proceed. We present the results in Table 3. We can see that in real interaction with a human user and allowing for clarification or correction, the models perform better compared to the rather strict automatic evaluation. Furthermore, the models are often successful in multiple sub dialogues, even if a part of the whole dialogue fails. The experiment also confirms the superior performance of ChatGPT on both dialogue success and JGA. Not surprisingly given the above results, conversations with ChatGPT also required fewer clarification turns than with Tk-Instruct.

6.2 Error Analysis

To understand the models’ behavior better, we manually inspect a random sample of ca. 20 dialogues for each model, chosen from cases where the automatic success metric was not satisfied. In general, we can split most of the erroneous behaviors into two distinct groups, which we call *prompt-recoverable* and *inherent*.

Prompt-recoverable errors can be likely fixed by specific prompt engineering with some effort. These errors happen with all of the tested models. Examples of such errors are the invalid structure of the generated dialogue state, copying slot values instead of using canonical values from the ontology, failure to delexicalize some of the values, etc. Most of these errors can be also fixed in postprocessing – for example, we can employ more robust parsers or fuzzy matching of slot values.

Inherent errors, on the other hand, are likely not easily fixable by prompt modifications. They are

	ChatGPT	Tk-Instruct
dialogues	25	25
subdialogues	52	48
clarify / dial	1.08	1.68
successful subdialogues	81%	71%
successful dialogues	76%	64%
correctly captured	88%	66%

Table 3: Human evaluation results for ChatGPT and Tk-Instruct-11B models. We evaluate the conversation on sub dialogue level i.e. each domain in the dialogue is evaluated separately.

not distributed evenly across the tested models and seem to constitute a more challenging problem.

Perhaps the most important error, common to all the models, is hallucination, i.e., the model’s output responses not grounded in the context (such as offering entities that are not included in the database). This happens in about 10-20% of the inspected dialogues. Some models (*GPT-NeoXT*, *OPT-IML*) tend to generate more content than they are asked for. This happens in more than 50% of their failed dialogues. In some cases, this means continuing the conversation for a few more turns (including hallucinating user turns), but the models also often generate unrelated text or even code snippets. With *Tk-Instruct*, we observed that in ca. 10% cases, it copies the belief state from the example given in the prompt instead of generating a relevant one. Another issue is that the models tend to repeat their previous responses.

7 Conclusion & Future Work

We present an experimental evaluation of instruction-tuned LLMs applied to the established task of task-oriented dialogue modeling, with five LLMs evaluated on two datasets. We find that LLMs are not performing well in terms of belief state tracking, even when provided with in-context few-shot examples. However, there is some potential to improve through prompt tuning and output parsing robust to irregularities.

If provided with a correct belief state, the models can interact with the user successfully, provide useful information and fulfill the user’s needs. While the performance does not match the supervised state of the art, it is important to note that these models were not finetuned on in-domain data and work with just a domain description or a few examples (which again improve performance).

Therefore, carefully picking representative ex-

amples and combining the LLM with an in-domain belief tracker can be a viable choice for a task-oriented dialogue pipeline.

Interestingly, in the human interactive evaluation, both ChatGPT and Tk-Instruct outperformed the expectations set by automatic metrics. This shows certain flexibility and ability to correct their own mistakes on the part of LLMs, and further demonstrates that single-turn evaluation is too rigid and does not show the whole picture (Takanobu et al., 2020). In future work, we want to focus on addressing the prompt-recoverable errors while maintaining the ability to use model-independent prompts and easily swap models. We also aim to find a more effective method of relevant example selection.

8 Limitations

One of the limitations of our work is the usage of the ChatGPT model, which is only accessible via an API and is not guaranteed to retain its exact abilities. However, the other four out of five evaluated models have publicly available weights and their results are fully reproducible. We still consider it beneficial to also evaluate ChatGPT, as it represents state-of-the-art at the time of writing of this paper and therefore puts the other models' results into perspective.

Another limitation is that based on our empirical experiments, the models are sensitive to the choice of a specific prompting. We spent some time finding a reasonably good prompt that would work with all of the models and did model-specific modifications for the evaluation. Specifically, the desired format of the belief state varied between the models, and there were some model-specific instructions. We also include both few-shot and zero-shot prompt types in our experiments. However, it is likely that the performance could be further improved with more extensive prompt engineering efforts. Nevertheless, we mainly aim to showcase the more raw/out-of-the-box capabilities of the LLMs, as extensive prompt tuning would, in practice, erase the advantage of not having to finetune the models. Furthermore, we believe that the robustness of the model to specific prompts also counts as an added value.

Finally, we cannot exclude the possibility that some of the models were exposed to our selected datasets during training. However, we still find it important to evaluate the LLMs in this setting.

Acknowledgments

This work was supported by the project TL05000236 *AI asistent pro žáky a učitele* co-financed by the Technological Agency of the Czech Republic within the ÉTA 5 Programme, by the European Research Council (Grant agreement No. 101039303 NG-NLG), and by the Charles University project SVV 260575. It used resources provided by the LINDAT/CLARIAH-CZ Research Infrastructure (Czech Ministry of Education, Youth and Sports project No. LM2018101).

References

- Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. 2022. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*.
- 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.
- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. *MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling*. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. ArXiv: 1810.00278.
- Deng Cai, Yan Wang, Wei Bi, Zhaopeng Tu, Xiaojiang Liu, and Shuming Shi. 2019. *Retrieval-guided dialogue response generation via a matching-to-generation framework*. 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 1866–1875, Hong Kong, China. Association for Computational Linguistics.
- Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.
- Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echevoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. 2021. *Survey on Evaluation Methods for Dialogue Systems*. *Artificial Intelligence Review*, 54:755–810.
- Yihao Feng, Shentao Yang, Shujian Zhang, Jianguo Zhang, Caiming Xiong, Mingyuan Zhou, and Huan Wang. 2023. Fantastic rewards and how to tame them: A case study on reward learning for task-oriented dialogue

- Yue Feng, Yang Wang, and Hang Li. 2021. [A sequence-to-sequence approach to dialogue state tracking](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1714–1725, Online. Association for Computational Linguistics.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
- Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A. Smith, and Mari Ostendorf. 2022. [In-context learning for few-shot dialogue state tracking](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2627–2643, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Tianjian Huang, Shaunak Ashish Halbe, Chinnadhurai Sankar, Pooyan Amini, Satwik Kottur, Alborz Geramifard, Meisam Razaviyayn, and Ahmad Beirami. 2023. [Robustness through data augmentation loss consistency](#). *Transactions on Machine Learning Research*.
- Xinting Huang, Jianzhong Qi, Yu Sun, and Rui Zhang. 2020. [MALA: cross-domain dialogue generation with action learning](#). In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7977–7984. AAAI Press.
- Chia-Chien Hung, Anne Lauscher, Ivan Vulić, Simone Ponzetto, and Goran Glavaš. 2022. [Multi2WOZ: A robust multilingual dataset and conversational pre-training for task-oriented dialog](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3687–3703, Seattle, United States. Association for Computational Linguistics.
- Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Dániel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, et al. 2022. Opt-impl: Scaling language model instruction meta learning through the lens of generalization. *arXiv preprint arXiv:2212.12017*.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.
- Jonáš Kulhánek, Vojtěch Hudeček, Tomáš Nekvinda, and Ondřej Dušek. 2021. [AuGPT: Auxiliary tasks and data augmentation for end-to-end dialogue with pre-trained language models](#). In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 198–210, Online. Association for Computational Linguistics.
- Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Seqicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447.
- Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- 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)*, pages 3391–3405, Online. Association for Computational Linguistics.
- Andrea Madotto, Zihan Liu, Zhaojiang Lin, and Pascale Fung. 2020. Language models as few-shot learner for task-oriented dialogue systems. *arXiv preprint arXiv:2008.06239*.
- 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.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.
- Gaurav Pandey, Danish Contractor, Vineet Kumar, and Sachindra Joshi. 2018. [Exemplar encoder-decoder for neural conversation generation](#). In *Proceedings of the 56th Annual Meeting of the Association for*

- Computational Linguistics (Volume 1: Long Papers)*, pages 1329–1338, Melbourne, Australia. Association for Computational Linguistics.
- 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.
- Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayan-deh, Lars Liden, and Jianfeng Gao. 2021. [Soloist: Building task bots at scale with transfer learning and machine teaching](#). *Transactions of the Association for Computational Linguistics*, 9:807–824.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- 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.
- Abhinav Rastogi, Raghav Gupta, and Dilek Hakkani-Tur. 2018. [Multi-task Learning for Joint Language Understanding and Dialogue State Tracking](#). In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 376–384, Melbourne, Australia. Association for Computational Linguistics.
- Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 05, pages 8689–8696.
- Nils Reimers and Iryna Gurevych. 2020. [Making monolingual sentence embeddings multilingual using knowledge distillation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- I. Shalyminov, A. Sordani, A. Atkinson, and H. Schulz. 2020. [Fast Domain Adaptation for Goal-Oriented Dialogue Using a Hybrid Generative-Retrieval Transformer](#). In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8039–8043. ISSN: 2379-190X.
- Igor Shalyminov, Sungjin Lee, Arash Eshghi, and Oliver Lemon. 2019. [Data-efficient goal-oriented conversation with dialogue knowledge transfer networks](#). 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 1741–1751, Hong Kong, China. Association for Computational Linguistics.
- Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. 2022. Selective annotation makes language models better few-shot learners. *arXiv preprint arXiv:2209.01975*.
- Haipeng Sun, Junwei Bao, Youzheng Wu, and Xiaodong He. 2022. Mars: Semantic-aware contrastive learning for end-to-end task-oriented dialog. *arXiv preprint arXiv:2210.08917*.
- Ryuichi Takanobu, Qi Zhu, Jinchao Li, Baolin Peng, Jianfeng Gao, and Minlie Huang. 2020. [Is Your Goal-Oriented Dialog Model Performing Really Well? Empirical Analysis of System-wise Evaluation](#). In *SIGdial*, pages 297–310, Online.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.
- Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022. Super-naturalinstructions: generalization via declarative instructions on 1600+ tasks. In *EMNLP*.
- Tsung-Hsien Wen, Milica Gašić, Dongho Kim, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. [Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking](#). In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 275–284, Prague, Czech Republic. Association for Computational Linguistics.

- Jason D. Williams, Kavosh Asadi, and Geoffrey Zweig. 2017. [Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 665–677, Vancouver, Canada. Association for Computational Linguistics.
- Yunyi Yang, Yunhao Li, and Xiaojun Quan. 2021. Ubar: Towards fully end-to-end task-oriented dialog system with gpt-2. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 16, pages 14230–14238.
- Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.
- 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*.
- Tiancheng Zhao and Maxine Eskenazi. 2018. [Zero-shot dialog generation with cross-domain latent actions](#). In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 1–10, Melbourne, Australia. Association for Computational Linguistics.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. [A Survey of Large Language Models](#). ArXiv:2303.18223 [cs].
- Qi Zhu, Christian Geishauser, Hsien chin Lin, Carel van Niekerk, Baolin Peng, Zheng Zhang, Michael Heck, Nurul Lubis, Dazhen Wan, Xiaochen Zhu, Jianfeng Gao, Milica Gašić, and Minlie Huang. 2022. [Convlab-3: A flexible dialogue system toolkit based on a unified data format](#). *arXiv preprint arXiv:2211.17148*.
- Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Prompt Construction

Prompt	<p>Determine which domain is considered in the following dialogue situation. Choose exactly one domain from this list: restaurant, hotel, attraction, taxi, train Answer with only one word, the selected domain from the list. You have to always select the most probable domain. —— Example 1: —— Customer: I need a cheap place to eat Assistant: We have several not expensive places available. What food are you interested in? Customer: Chinese food. Domain: restaurant —— Example 2: —— Customer: What is the address? Assistant: It's 123 Northfolk Road. Customer: That's all. I also need a train from London. Domain: train —— Now complete the following example: Customer: I am looking for a cheap place to stay. Domain:</p>
Output:	hotel

Table 4: A prompt used for domain detection for MultiWOZ. It contains **task definition**, **domains description**, **static examples** and **user utterance**.

Prompt	<p>Definition: Capture entity values from last utterance of the conversation according to examples. Capture pair "entity:value" separated by colon and no spaces in between. Separate entity:value pairs by hyphens. If not specified, leave the value empty. Values that should be captured are: - "pricerange": the price of the hotel - "area" that specifies the area where the hotel is located (north/east/west/south/centre) - "internet" that specifies if the hotel has internet (yes/no) - "parking" that specifies if the hotel has parking (yes/no) - "stars" that specifies the number of stars the hotel has (1/2/3/4/5) - "type" that specifies the type of the hotel (hotel/bed and breakfast/guest house) [history] Customer: "I want a cheap place to stay."</p>
Output:	pricerange:"cheap"

Table 5: A zero-shot version of the prompt used for state update prediction for MultiWOZ 2.2. It contains **task definition**, **domain description**, **dialogue history** and **user utterance**.

Prompt	<p>Definition: You are an assistant that helps people to book a hotel. The user can ask for a hotel by name, area, parking, internet availability, or price. There is also a number of hotel in the database currently corresponding to the user's request. If you find a hotel, provide [hotel_name], [hotel_address], [hotel_phone] or [hotel_postcode] Do not provide real entities in the response! Just provide entity name in brackets, like [name] or [address]. If booking, provide [reference] in the answer. [history] Customer: "I want a cheap place to stay." State: hotel { pricerange: "cheap"} Database: hotels: 23</p>
Output:	We have 23 such hotels available, do you have a preference about the location?

Table 6: A zero-shot version of the prompt used for response prediction for MultiWOZ 2.2. It contains **task definition**, **domain description**, **dialogue history**, **user utterance** and **belief state with db results**.