Did You Get That? Evaluating GPT-4's Ability to Identify Additional Context

Anonymous ACL submission

Abstract

In recent years, Large language models (LLMs) have emerged as powerful knowledge bases. Despite increasing adoption, little is known about their true capabilities. We evaluate the strengths and weaknesses of the state-of-the-art in LLMs when identifying additional context in dialogue. We define additional context as information supplied by the user that is not directly asked of them. We specifically evaluate GPT-4 and its ability to recognize such information. While GPT-4 can accurately identify additional information in some sentences, it fails to identify additional context more than 22% of the time. By understanding these limitations, we can remain aware of pitfalls and harness LLMs within the scope of their abilities.

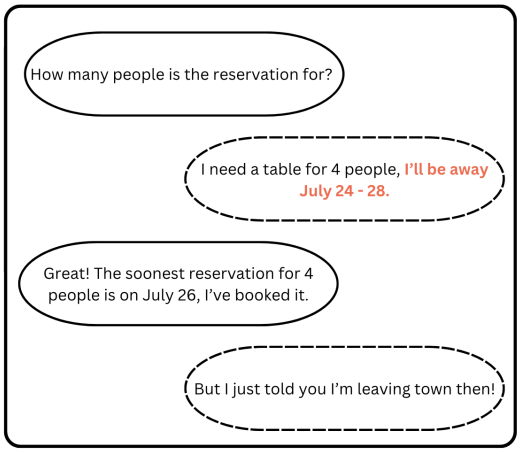

Figure 1: An example conversation between an agent (solid line) and a user (dashed line). The user's first response includes additional context, highlighted in red.

1 Introduction

Large language models (LLMs) are tools capable of producing believable text and perform well on a variety of tasks. As the field of natural language processing develops, the adoption of LLMs has become more widespread. One existing application of LLMs is in the use of dialogue agents, which are agents that converse with a human user. Use cases for dialogue agents are vast and constantly evolving (Teixeira and Dragoni, 2022). Automated technical support, online customer service, and reservation booking systems are just a few of many uses for dialogue agents.

In the dialogue setting, we define additional user context as information provided by the user that is important to retain but is not the response to the question asked. In Figure 1, we can see the dialogue agent query the user, asking how many people they need a reservation for. The user provides the answer to the question, but they also provide a bit more – they've indicated dates that they will be out of town. What we refer to as additional context is in red. While it might seem trivial for humans to retain this information, this is not the case for many structured dialogue agents.

Retaining additional context in the case of dialogue agents is particularly important for goal-oriented conversations. Goal-oriented conversations have, as the name suggests, a goal that is trying to be achieved during the conversation. Resolving an issue with online technical support is an example of a goal-oriented conversation – that is, there is an outcome of the conversation that needs to be satisfied. To achieve such goals, detailed information is required from an individual. This requires a turn-based conversation with question-answering to extract the required information. If a dialogue system ignores additional context from the user, they have the potential to miss out on valuable information. Additionally, ignoring this information could make conversations longer if they are seeking information by asking another question when the answer was already supplied earlier on in the conversation. Disregarding this additional context, as is evident further on in the conversation in Figure 1, can lead to conversation goals not being resolved.

Ultimately, users can become dissatisfied, feel like they aren't being heard, and have a generally poor experience if we do not pay attention to additional context. For this reason, we explore the current capabilities of LLMs at identifying additional context in dialogue settings.

While this may seem like an inconsequential example, as we see dialogue agents becoming more popular in detail-oriented domains where customer trust and satisfaction is paramount, such a banking, it is imperative that we understand if, how, and when LLMs fail. Therefore, we have conducted this work seeking to understand how well the current state-of-the-art LLM can identify and extract the additional user context. Concretely, we explore three research questions:

- **RQ1:** Can GPT-4 extract additional context from a question-answer pair?
- **RQ2:** How does zero-shot, one-shot and few-shot prompting impact model performance?
- **RQ3:** If additional context can be extracted, can values be extracted from the result?

To answer these research questions, we use the current state-of-the-art language model, OpenAI's GPT-4 (OpenAI, 2023), and evaluate how it performs on a variety of additional context extractions. As we aim to explore its current abilities, we forgo any fine-tuning and only perform zero-shot training tasks. Based on the Multi-WOZ (Eric et al., 2019) dataset, we have constructed a testing dataset containing agent questions and user responses with varying sentence structures and degrees of additional context. We explore the impacts of zero-shot, one-shot, and few-shot prompting on GPT-4 using this testing dataset. As a follow-up task, we evaluate how GPT-4 performs at assigning the identified information to slots (variables). Overall, we found that GPT-4 performs well on simple tasks, but as language becomes more nuanced, difficulties arise in accurately identifying which information was offered as an answer to the question and which information is provided as additional context. In summary, our work makes the following contributions:

- A testing dataset containing query-response text with labelled additional user context.
- An analysis of how GPT-4 performs at additional context extraction with zero-shot training and different shot prompting schemes.

2 Background & Notation

2.1 Dialogue Agents

Dialogue agents are computer systems designed to converse with a human in natural language. Goal-oriented dialogue systems (sometimes referred to as task-oriented) are dialogue systems that look to achieve a specific outcome (Jurafsky and Martin, 2017). Turn-based dialogue is used to fill slot values that are relevant to achieving a goal. Slots can be thought of as variables that need values assigned to them. For example, the goal of booking a hotel could require slot values for dates, number of guests, room size, and location. Possible slot values for the number of guests might be any number from one to five.

2.2 Large Language Models

LLMs are transformer-based models that can generate natural language. Recent years have shown significant growth in LLMs and their use cases. In the context of dialogue systems, LLMs are used as a natural language generation component (Teixeira and Dragoni, 2022). While a number of LLMs exist, the current state-of-the-art is GPT-4 from OpenAI (OpenAI, 2023). While GPT-4 performs well on some human tasks, it still suffers from the same issue as its predecessors: it can make up or 'hallucinate' information (OpenAI, 2023). Prompting is used to generate output from LLMs. *X-Shot* in terms of prompting refers to the number of examples given before the task. So a one-shot task includes one example in the prompt.

2.3 MultiWOZ Dataset

The MultiWoz dataset is freely available under the MIT license and contains task-oriented, human-human conversations (Budzianowski et al., 2018). The dataset contains seven domains: Attraction, Hospital, Police, Hotel, Restaurant, Taxi, and Train (Budzianowski et al., 2018). These domains contain a total of 25 slots to fill, as well as a combined total of 4510 potential slot values (Budzianowski et al., 2018). The authors provide three example use cases of dialogue state tracking, dialogue-context-to-text generation, and dialogue-act-to-text generation, however these are not the only use cases (Budzianowski et al., 2018). This dataset has seen multiple iterations (Ramadan et al., 2018; Zang et al., 2020), and our research builds on MultiWOZ 2.1 (Eric et al., 2019).

3 Approach

To answer our research questions, we first construct a dataset of question-answer pairs across three single domain and three dual domain tasks. Dual domain tasks include two topics, rather than just one in the single domain. Using this data, we then look to achieve two tasks. In Task 1, we look to extract additional context from the conversation. In Task 2, we use the extracted information from Task 1 to fill slot values.

3.1 Dataset

We constructed a preliminary dataset of questions and responses that include varying degrees of additional user context across multiple domains. We base our domains and slot values on the MutliWOZ 2.1 dataset (Eric et al., 2019). The following section outlines the domains used, sentence templates, and the aggregating of the final dataset.

3.1.1 Domains

We divided our templates into two domain types: single domain and dual domain. The single domain includes three domains: Hotel, Restaurant, and Train. These domains were selected from the 2.1 version of the MultiWOZ dataset (Eric et al., 2019) as they center around a similar topic (travel), and because they are approachable to most users. Restaurant dialogue tasks have the goal of booking a reservation at based on preferences. Hotel dialogue tasks focus on booking a hotel based on preferences. Finally, Train dialogue tasks look to book train travel based on user preferences. Each domain has a set of associated slots that are relevant to achieving its goal. Table 1 summarizes the slots for each individual domain. Similar to the domains themselves, these slots were adapted from the MultiWOZ 2.1 dataset (Eric et al., 2019). Explained further in Section 3.1.2, each slot also has possible values it can take on from the same dataset (Eric et al., 2019).

The dual domain templates contain pairings of the domains from the single domain category: Hotel-Train, Restaurant-Hotel, and Restaurant-Train. The dual domain goals are a combination of both single domain goals. For example, the Restaurant-Hotel domain has the goal of booking both a restaurant and hotel based on user wants. Similarly, each dual domain's slots are a combination of the single domain's slots. So, Hotel-Train's slots are the union of Hotel slots and Train slots.

Domain	Slots
Hotel	hotel_area hotel_book_day hotel_name hotel_pricerange hotel_people hotel_stars hotel_stay
Restaurant	restaurant_booking_time restaurant_food_types restaurant_day restaurant_names restaurant_people
Train	train_arrive_by train_day train_departure train_destination train_leave_at train_people

Table 1: Single domains with their associated slot values. For dual domain slots, combine both lists.

In total, we have six domains spanning two domain types: Hotel, Restaurant, Train, Hotel-Train, Restaurant-Hotel, and Restaurant-Train.

3.1.2 Templates

For each of the six domains outlined above, we construct ten template sentences. Each of the ten templates is a user's response to a direct question. In constructing the templates for the single domain domains, we focused on three different cases of additional context appearing in these templates. In the first (base) case, the user answers the question directly and does not provide additional context. In the second case, the user answers the question directly and provides a single piece of additional context. In the third case, the user does not answer the question directly but does provide a single piece of additional context.

In constructing the templates for the dual domain domains, we ignore the notion of no context or all context as this is captured in the single domain experiments. Here, we are more concerned with how combining two different domains can impact the distinction between main information and additional context. In all dual domain templates, the main information and additional context belonged to different domains.

When creating templates for all six domains, we also varied the number of sentences and where the additional context appears in the sentence. Some templates are composed of two sentences: one for the main information and one for the additional context. Other templates are composed of only one sen-

Question	Template	# Sentences	Reverse?
How many people is the reservation for?	I need to book a table for *restaurant_people*, I'll be available *restaurant_day*.	1	no
What type of cuisine would you like to eat?	I'd like to eat some *restaurant_food_types* cuisine. I'd also like to check out *restaurant_names* while I'm here.	2	no
When would you like to book your reservation for?	*restaurant_names* is our usual hangout, but let's change up the time to *restaurant_day*.	1	yes
What day would you like to book the reservation on?	We're actually going to be *restaurant_people* people now. Let's book for *restaurant_day*.	2	yes
What time would you like to dine?	Normally I eat at *restaurant_names*.	no main, just context	
When would you like to book your reservation?	I'd like to eat at *restaurant_booking_time*.	just main, no context	

Table 2: Sample templates with their associated question from the Restaurant domain. Slot names are italicized.

tence where the main information and additional context are (optionally) joined by a connective. To determine if the order in which information appears affects GPT-4's ability to discern additional context from the main information, we vary whether the main information comes before or after the additional context. Table 2 shows templates of each of the described types.

We chose to create templates rather than complete sentences to control for slot values impacting results. If we use multiple templates filled with different slot values, we can be more confident that the sentence structure is impacting results rather than the specific slot value. For each template, the slots were filled in five different ways to give five different sentence variants. To fill sentences, each template contains slot names as placeholders. These slot names are then matched to a list of variable names present in the MultiWOZ dataset. We ensured no identifying details were present in the slot values and removed any values listed as 'don't care'. A random value is selected for each slot in the template. Figure 2 shows an example of a template and two possible filled variations.

Template: The *train_people* of us want to go to *train_destination*.

Filled-1: The *9* of us want to go to *Leicester*.

Filled-2: The *2* of us want to go to *Broxbourne*.

Figure 2: A sample template from the train domain with two possible, randomly selected slot values.

3.1.3 Final Dataset

The final dataset consists of 300 question responses using 60 templates spread evenly across the six domains. The initial question and slot values are recorded, and ground truths for the main information and additional context are included. The number of sentences (1 or 2) used in a template is recorded, as well as whether the additional context comes before or after the main information (reversed or not reversed). As we are not conducting any training or fine-tuning, a dataset of 300 data points is appropriate, and we need not worry about keeping training and testing data separate.

We constructed the dataset in Python using `Pandas` (Wes McKinney, 2010; pandas development team, 2020) and built-in string functions. We felt it was important that the dataset is structured so that it can remain dynamic for future research. Adding additional templates for each domain is as simple as appending a line to a master template spreadsheet, and from there, you can repopulate the slots with a script. This allows for additional interesting use cases to be added as templates for further experimentation, as this dataset is not an exhaustive list. All materials used for this work, including dataset, prompts, code, and documentation, are available on GitHub at `https://github.com/`[1].

3.2 Task 1: Extracting Additional Context

For this task, GPT-4 is given a question-answer conversation from our dataset. It is asked to extract what was the main information queried from the user and what was added as additional context.

In our work, we experimented with a variety of prompts to perform zero-shot, one-shot, and few-shot extraction. A key feature of our prompt exploration is ensuring that the answer produced is a direct subset of the dialogue given. That is,

[1]Repository name has been redacted to maintain author anonymity.

GPT-4 may not paraphrase the answer and must provide a direct quote. This ensures answers are verifiable and attempts to combat a well-known issue with LLMs: their tendency to hallucinate (OpenAI, 2023).

For zero-shot extraction, the task is outlined, and the conversation is given. No examples are provided. For both one-shot and few-shot, the task is outlined, and the conversation is given, but examples are also included. These examples show a handful of ways that additional context may be present and extracted. We picked a domain for this example that was independent of our domains to avoid skewing results in favour of one domain or another. Figure 3 shows an example of one of the three prompts used. zero-shot prompting is identical to one-shot but does not include examples. It does, however, include instructions for answer formatting. few-shot prompting is identical one-shot prompting but includes two additional examples. All prompts are given in Appendix A.

One-Shot Prompt

You will be given a conversation between a User and an Agent. The Agent will ask a direct question and the User answers the question. The User may also add extra context that wasn't asked of them, or they may not. Your task is to identify the Main Information and the Additional Context. These must be direct quotes from the conversation and have no additional text added. If there is no Additional Context found, you can label the Additional Context as 'None'. One example is provided below.

EXAMPLE 1
Agent: How much money would you like to deposit?
User: I need to deposit $100 and I'd also like to pay off my credit card bill.

Main Information: I need to deposit $100
Additional Context: I'd also like to pay off my credit card bill.

Your conversation is:

<conversation>

Figure 3: One shot prompting used to extract additional context.

3.3 Task 2: Slot Assignment

The second task was created to determine if GPT-4 is capable of assigning the main information and additional context to the slot values present. The second task follows the first and is dependent on its results. There are two cases for the results: either the main information and additional context were extracted correctly, or they were not.

In the case that GPT-4 successfully completed Task 1, the result was passed to GPT-4 with the possible slots and asked to map the available information to matching slots. In the case that GPT-4 did not complete Task 1 successfully, the correct answer was manually produced from ground truth and then passed to GPT-4 with the possible slots and asked to map the information to the slots. In both cases, only one filled instance from each of the templates is selected. Without loss of generality, we select the first filled instance of each template, giving a testing set of 60 instances. The prompt used is included in Figure 4, and it intentionally did not mention that there were more slots than information presented.

Follow Up Prompt

Now that you have identified the Main Information and Additional Context, you must assign each of the Main Information and the Additional Context to a named variable below.

<possible slot 1>
<...>
<possible slot n>

To recall, you identified:

<Main Information>
<Additional Context>

Figure 4: The follow-up prompt used in Task 2 for slot filling.

4 Evaluations

Our experiments were conducted on a machine running Ubuntu 22 with 8 CPU cores, 300GB of RAM, and a single NVidia A40 GPU accelerator. In Task 1 and Task 2, GPT-4 was used with the default parameters (temperature=1, max_tokens=256, top_p=1, frequency_penalty=0, presence_penalty=0). Following Task 1, each instance was manually observed and classified as correct or incorrect. Results were aggregated first by single domain or dual domain and then further sorted by individual domain. We grouped templates where at least one instance was incorrect. Finally, we manually classified correct or incorrect results from Task 2.

4.1 Results

To gain insights into how GPT-4 performs at extracting additional context, we use our results from

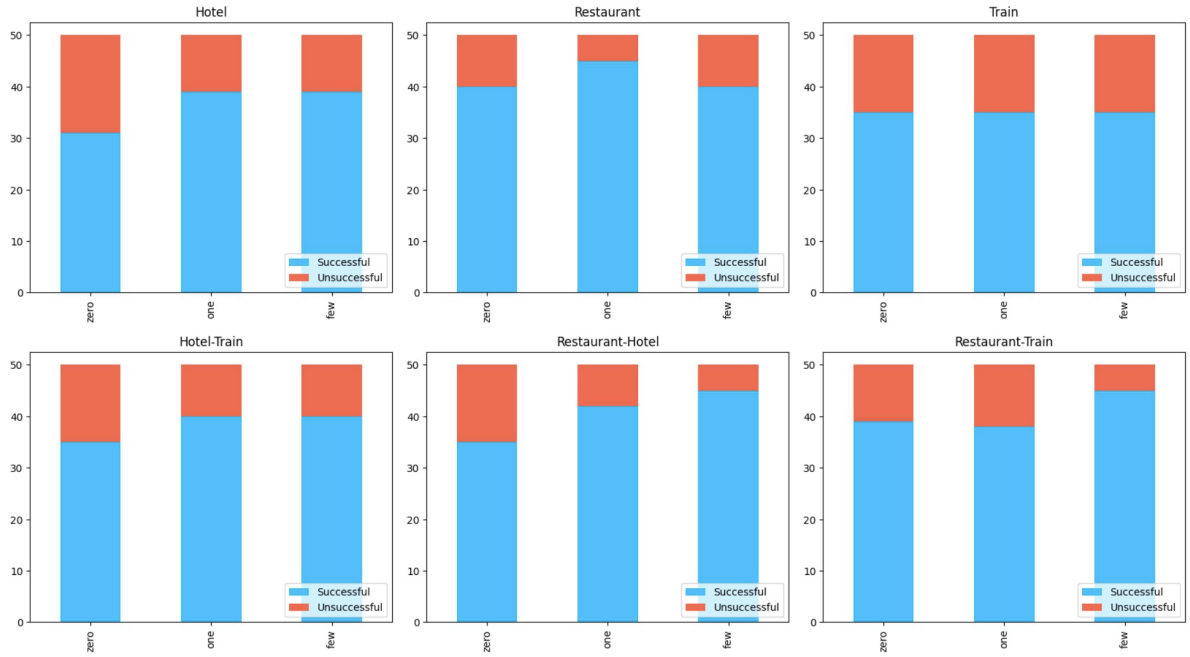

Figure 5: Number of successful versus unsuccessful additional context extractions for each prompting technique (zero-shot, one-shot, few-shot), separated by domain.

Task 1 and Task 2 performed on our dataset to answer our research questions below.

RQ1: Can GPT-4 extract additional context from a question-answer pair?

To answer RQ1, we look to the results from Task 1. Figure 5 summarizes the results across the six domains. In the worst-case scenario (Hotel, zero-shot), the additional context was only successfully extracted in 31 of 50 instances or 62% of the time. In the best case scenario(s) (Restaurant one-shot, Restaurant-Hotel one-shot, and Restaurant-Train one-shot), the additional context was successfully extracted in 45 of 50 instances or 90%. Overall, across the six domains and three different prompting styles, the additional context was identified successfully 698 of 900 instances[2], or 77.56%.

We then view our results from the perspective of templates rather than individual-filled instances. In both single domain and dual domain, there were 21 template types (of 30 total) where additional context was identified 100% of the time by all three prompting types. There were nine templates where at least one instance was incorrect.

In Table 3, we have summarized all templates that had at least one instance where additional context could not be identified (or was misidentified). As expected, we do not see any single domain base cases in Table 3 where no additional context was offered, just the direct answer to the question. This reinforces that GPT-4 is able to identify the main information in a sentence successfully.

Working down the list in Table 3, the first three templates are all missing main information and are just additional context. These are the only three templates of this type, that is, a 0% success rate when the question topic is ignored. This suggests difficulties in cases where the question wasn't answered by the user, or that prompting with this type of example should be included.

We can see from the second column in Table 3 that the majority of templates that were not successfully identified were composed of a single sentence. Another pattern that emerges is that a large number of templates are reversed; that is, the additional context comes before the main information. Looking at the combination of the two, we see most of misclassifications happen when it is a single, reversed sentence. Of the 60 templates, there are 16 single, reversed sentences. In the misclassified sentences in Table 3 there are 10. While GPT-4 does not misclassify additional context in every single, reversed sentence, the results suggest that this style of sentence poses the greatest challenge.

Overall, our results suggest that while GPT-4 can extract additional context successfully, it struggles

[2]300 filled templates multiplied by three outputs (one each for zero-shot, one-shot, and few-shot).

Template	Number of Sentences	Reversed?	# Correct		
			Zero-Shot	One-Shot	Few-Shot
We need to go on *train_day*.	1	no	0	0	0
Normally I eat at *restaurant_names*.	1	no	0	0	0
There'll be *hotel_people* of us.	1	no	0	0	0
train_people of us need to get to *hotel_name*.	1	no	0	0	0
train_people of us will be traveling there and *restaurant_people* of us will be eating.	1	no	0	0	0
We need to be there by *train_arrive_by* on *train_day*.	1	yes	0	0	0
The *train_people* of us want to go to *train_destination*.	1	yes	0	0	0
We're looking for something *hotel_pricerange* for *hotel_stay* nights.	1	yes	0	0	0
We need to check-in on *hotel_book_day*, let's book for *restaurant_day*.	1	yes	0	0	0
We want to stay for *hotel_stay* days and we need tickets for *train_day*.	1	yes	0	0	0
I'm craving *restaurant_food_types* so let's head to *train_departure*.	1	yes	0	0	4
Since we'd like to eat at *restaurant_names* I'd prefer the *hotel_area* area of the city.	1	yes	0	2	5
We're getting here *hotel_book_day* and staying for *hotel_stay* days.	1	no	0	4	4
My go to spot is *restaurant_names*, but it's closed. How about we try *restaurant_food_types* cuisine.	2	no	1	4	0
We check-in on *hotel_book_day*. Let's arrive by *train_arrive_by*.	2	yes	1	4	5
We'd like to stay in the *hotel_areas* part of the city, so a reservation at *restaurant_names* would be nice.	1	yes	2	5	5
Since I'd like to stay at the hotel_name, I'm looking for something *hotel_pricerange*.	1	yes	4	3	5
To make our dinner reservation at *restaurant_booking_time* I want to arrive by *train_arrive_by*.	1	yes	4	3	5

Table 3: All templates that were misclassified by GPT-4 at least once.

with certain types of sentences more than others. In particular, single sentences, when the additional context appears before the main information. Difficulties also arise when the question is ignored, but other information is offered up by the user. We explore how prompting affects these results below.

RQ2: How does zero-shot, one-shot, and few-shot prompting impact model performance?

By observing the graphs shown in Figure 5, we can begin to understand how prompting affects performance to answer RQ2. There are differences between zero-shot, one-shot and few-shot in all domains except for Train. When you observe individual data points, in most cases, it is consistent across all prompting shots. I.e., either all correct or all incorrect. Indeed, only 8 of 60 templates showed any variation between prompting types. These are all shown in the later half of Table 3.

Overall, in the 40 [3] instances where there were differences in results based on prompting , zero-shot prompting produced correct results 12 times (30%), one-shot produced correct results 25 times (62.5%), and few-shot produced correct results 33 times (82.5%). There is a clear indication that more examples increases performance for these 8 instances. This suggests that robust examples are necessary to exact additional context in some instances.

RQ3: If additional context can be extracted, can values be extracted from the result?

Finally, we look to the results of Task 2 to answer RQ3. Table 4 shows the aggregated results. Overall, only 12 of the 60 instances were incorrect, or a 20% failure rate. Interestingly, in the single domain, all 21 templates that were correctly classified in Task 1, were also correctly assigned to variable names in Task 2. This was not the case for successful templates in the dual domain, where only 13 of 21 had correct variable assignments. The most common error was attributing value to the wrong

[3] 8 templates * 5 variations = 40 instances

	Type	Right Task 2	Wrong Task 2	Total
Right Task 1	**Single**	21	0	21
	Dual	13	8	21
Wrong Task 1	**Single**	7	2	9
	Dual	7	2	9
Total		**48**	**12**	**60**

Table 4: Results for Task 2, based on success in Task 1, separated by either single domain or dual domain type.

domain. For example, Hotel, Restaurant, and Train all have a variable for the number of people.

For both single domain and dual domain, the unsuccessful templates from Task 1, were still able to have the correct variable assignment the majority of the time, given that the correct main information and additional context were provided. Based on the results from RQ1, we are curious if sentence structure may play a role here. In general, we can say that given correct answers from Task 1, GPT-4 can assign slot values from a sentence. However, its overall success rate is 80%.

4.2 Discussion

Overall, we observed that GPT-4 has some proficiency in extracting additional context from question-answer conversations. However, it is limited. Single sentences, especially when the additional context appears before the main information, are most troublesome. GPT-4 is also capable of assigning portions of the sentence to fill slot values based on extracted main information and additional context. Results from RQ1 and RQ2 also suggest that increasing the type and number of prompts used can impact performance. However, this leads to issues in collecting a large number of examples with coverage of the response space. Issues arise also when assigning variables to more than one domain at once. We expect similar results if we extend the number of domain combinations.

We are operating in a space where assumptions have largely constrained possible sentences. We assume that, at most, one piece of additional information is added. This may not always be the case for real human behaviour. We have also not accounted for wider variation in English fluency, formality, and brevity (or verbosity) in these sentences. Lastly, we have assumed a small number of slots per task; more complex real-world tasks could contain more slots to fill and relaxing these assumptions could lead to a decrease in performance.

5 Related Work

Context is a familiar notion in dialogue agents. However, its definition and use differ from our work. We see the term context crop up for background information on a task, or existing knowledge of a user (Wei et al., 2018; Suresh et al., 2022; Guo et al., 2017). Our work narrows the notion of context to other information required from the goal that is offered by the agent in response to a query. Extracting additional context could, in turn, expand the existing context base for a given user. Larson et al. look at implicit information given by a user in conversation and its privacy concerns (2021). This differs from our work, as we explore information explicitly offered by the user that is relevant to the goal. Perhaps the most related to our work, OrchestraLLM is a routing framework combining task-specific small language models with an LLM to outperform LLM-only based approaches on task-oriented dialogue (Lee et al., 2023). This differs from our approach as the authors do not specifically consider how additional information may be handled.

6 Summary

6.1 Conclusions

In this work, we explored how GPT-4 performs at extracting additional context in question-answer pairs. We have developed a preliminary dataset and prompted GPT-4 to both extract additional context and assign slot values. We found that while additional context can be extracted with 77.56% accuracy, when the direct question is not answered or in single sentences where the additional information comes before the answer, GPT-4 struggles. Slot filling tasks can be completed with 80% accuracy. As we move from the travel sector into more sensitive domains, our tolerance for error decreases. If individuals or companies choose to use LLMs in chatbots, we must be aware of the risks of models such as GPT-4. While general success has been shown, we risk losing information in the other 22.44% of cases. Missing important additional context in the medical sector, for example, could have significant consequences.

6.2 Ethical Considerations and Broader Impact

Any research surrounding LLMs must take into account the ethical concerns of using such models. While some may see this work as a reason to use

LLMs, given greater than 75% accuracy, our goal is to present a cautious lens over the use of LLMs in the dialogue setting.

Extracting additional information that does not relate to the topic at hand can have security implications. Larson et al. discuss in their work the possible privacy threats that come when implicit information is extracted from users, especially when users are unaware that additional information that they have given is relevant (2021). In our work, we focus explicitly on additional context that is still relevant to dialogue-goal. However, it is possible that understanding how additional context can be extracted by LLMs (at least with some success) could lead to malicious activity and extracting implicit information the user never intended to provide. Finally as with all research conducted on or using LLMs, we are subject to the inherent bias in the scraped internet data used for GPT-4's training (OpenAI, 2023). As we are using a black box model, it is unclear how bias in this model has affected this work, or future extensions or applications of this work.

7 Limitations

With increasing use and hype surrounding the use of LLMs, we believe it is important to understand where their performance suffers. Our goal in this work is to establish a preliminary baseline in the performance of GPT-4 on additional context extraction. However, we understand that a number of limitations exist in our work. By understanding our limitations, we hope that we, as well as other researchers, can build upon these results to develop a more concrete picture of how LLMs perform on a wide range of goal-oriented dialogue tasks.

An obvious limitation of our work is that it has been conducted only in English by two English first-language authors. The idea of additional context extends beyond English and results from applying a similar task in other languages would be a huge improvement that we are not properly equipped to take on. This also calls into question ethical concerns about the types of training data used for large language models. Tasks performed well in English may not be performed well in other languages that make up a smaller proportion of the training dataset used for GPT-4. Exploring this task in various languages could shine a light on the language bias inherent in LLMs.

In a similar vein, our templates used colloquial English, again developed by researchers whose first language is English. We do not fully explore how different fluency, vernacular, and writing patterns can impact results. Differences in spelling and formality have not been taken into consideration in this work and could impact the results. Some users may be overly brief when conversing with dialogue agents; this has also not been incorporated in the templates.

The templates created are not an exhaustive list of possible sentence structures that include additional context. While we have a basic knowledge of connectives and clause structure, consultation with an expert in syntax could provide more template possibilities that have not initially been considered. Knowing that we do not have a complete list of possible sentences, we have factored this limitation into our work by structuring our code so that new templates can be easily added and tested.

We have also not compared our results across other LLMs. While GPT-4 is considered the current state of the art, comparison to other LLMs would provide a more robust picture of the current landscape of LLM ability.

References

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Ultes Stefan, Ramadan Osman, 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 (EMNLP)*.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.

Xiaoxiao Guo, Tim Klinger, Clemens Rosenbaum, Joseph P. Bigus, Murray Campbell, Ban Kawas, Kartik Talamadupula, Gerry Tesauro, and Satinder Singh. 2017. Learning to query, reason, and answer questions on ambiguous texts. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Daniel Jurafsky and James H Martin. 2017. Dialog systems and chatbots. *Speech and language processing*, 3.

Martha A. Larson, Nelleke Oostdijk, and Frederik J. Zuiderveen Borgesius. 2021. Not directly stated, not explicitly stored: : Conversational agents and the privacy threat of implicit information. In *Adjunct*

Publication of the 29th ACM Conference on User Modeling, Adaptation and Personalization, UMAP 2021, Utrecht, The Netherlands, June 21-25, 2021, pages 388–391. ACM.

Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. 2023. Orchestrallm: Efficient orchestration of language models for dialogue state tracking. *CoRR*, abs/2311.09758.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

The pandas development team. 2020. pandas-dev/pandas: Pandas.

Osman Ramadan, Paweł Budzianowski, and Milica Gasic. 2018. Large-scale multi-domain belief tracking with knowledge sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 432–437.

Abhijit Suresh, Jennifer Jacobs, Margaret Perkoff, James H Martin, and Tamara Sumner. 2022. Fine-tuning transformers with additional context to classify discursive moves in mathematics classrooms. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 71–81.

Milene Santos Teixeira and Mauro Dragoni. 2022. A review of plan-based approaches for dialogue management. *Cogn. Comput.*, 14(3):1019–1038.

Wei Wei, Quoc V. Le, Andrew M. Dai, and Jia Li. 2018. Airdialogue: An environment for goal-oriented dialogue research. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3844–3854. Association for Computational Linguistics.

Wes McKinney. 2010. Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference*, pages 56 – 61.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, 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, ACL 2020*, pages 109–117.

A Appendix A

The following sections include the full prompts used as well as all template sentences.

A.1 Prompts

Zero-Shot Prompt

You will be given a conversation between a User and an Agent. The Agent will ask a direct question and the User answers the question. The User may also add extra context that wasn't asked of them, or they may not. Your task is to identify the Main Information and the Additional Context. These must be direct quotes from the conversation and have no additional text added. If there is no Additional Context found, you can label the Additional Context as 'None'.

Your answer should look like:

Main Information: ...
Additional Context ...

Your conversation is:

<conversation>

Figure 6: Zero shot prompt used.

One-Shot Prompt

You will be given a conversation between a User and an Agent. The Agent will ask a direct question and the User answers the question. The User may also add extra context that wasn't asked of them, or they may not. Your task is to identify the Main Information and the Additional Context. These must be direct quotes from the conversation and have no additional text added. If there is no Additional Context found, you can label the Additional Context as 'None'. One example is provided below.

EXAMPLE 1
Agent: How much money would you like to deposit?
User: I need to deposit $100 and I'd also like to pay off my credit card bill.

Main Information: I need to deposit $100
Additional Context: I'd also like to pay off my credit card bill.

Your conversation is:

<conversation>

Figure 7: One shot prompt used.

Few-Shot Prompt

You will be given a conversation between a User and an Agent. The Agent will ask a direct question and the User answers the question. The User may also add extra context that wasn't asked of them, or they may not. Your task is to identify the Main Information and the Additional Context. These must be direct quotes from the conversation and have no additional text added. If there is no Additional Context found, you can label the Additional Context as 'None'. One example is provided below.

EXAMPLE 1
Agent: How much money would you like to deposit?
User: I need to deposit $100 and I'd also like to pay off my credit card bill.

Main Information: I need to deposit $100
Additional Context: I'd also like to pay off my credit card bill.

EXAMPLE 2
Agent: What type of bank account would you like to open?
User: I just retired, so I think it'd be best to open a savings account.

Main Information: it'd be best to open a savings account.
Additional Context: I just retired

EXAMPLE 3
Agent: Who would you like to authorize on your account?
User: I'd like to authorize my husband. He'll also need to open a new savings account.

Main Information: I'd like to authorize my husband.
Additional Context: He'll also need to open a new savings account.

Your conversation is:

<conversation>

Figure 8: Few shot prompt used.

Follow Up Prompt

Now that you have identified the Main Information and Additional Context, you must assign each of the Main Information and the Additional Context to a named variable below.

<possible slot 1>
<...>
<possible slot n>

To recall, you identified:

<Main Information>
<Additional Context>

Figure 9: Follow up prompt used

A.2 Templates

Tables 5 and 6 on the following page show the templates for the single domain and the dual domain respectively. The second column gives the question asked and the template in column three is the response. Column four shows the number of sentences and column five indicates whether or not it is reversed.

Domain	Question	Template	# Sentences	Reverse?
Restaurant	How many people is the reservation for?	I need to book a table for restaurant_people, I'll be available restaurant_day.	1	no
Restaurant	What type of cuisine would you like to eat?	I'd like to eat some restaurant_food_types cuisine. I'd also like to check out restaurant_names while I'm here.	2	no
Restaurant	When would you like to book your reservation for?	We're actually going to be restaurant_people people now. Let's book for restaurant_day.	2	yes
Restaurant	What day would you like to book the reservation on?	restaurant_names is our usual hangout, but let's change up the time to restaurant_day.	1	yes
Restaurant	Do you have a specific restaurant in mind?	My go to spot is restaurant_names, but it's closed. How about we try restaurant_food_types cuisine.	2	no
Restaurant	How many people is the reservation for?	I need to book a table for restaurant_people, I'll be leaving town restaurant_day.	1	no
Restaurant	What type of cuisine would you like to eat?	I was hoping to eat some traditional restaurant_food_types cuisine while I'm here. I've heard restaurant_names is great.	2	no
Restaurant	What time would you like to dine?	Normally I eat at restaurant_names.	1	no
Restaurant	What type of cuisine do you like?	I'm dining with a group of restaurant_people tonight. Depending on who I'm with, I normally go for restaurant_food_types.	2	yes
Restaurant	When would you like to book your reservation?	I'd like to eat at restaurant_booking_time.	1	no
Hotel	How many people will be staying?	There's going to be hotel-people of us arriving on hotel_book_day.	1	no
Hotel	What day will you arrive?	We want to show up on hotel_book_day. I think we need to book for hotel_stay nights.	2	no
Hotel	Do you have a preference for the area the hotel is in?	Because we'll be staying for hotel_stay, I'd prefer to be in the hotel area.	1	yes
Hotel	How long are you planning on staying for?	We're looking for something hotel_pricerange for hotel_stay nights.	1	yes
Hotel	Do you have a specific hotel in mind?	There'll be hotel-people of us.	1	no
Hotel	What is your pricerange?	Since I'd like to stay at the hotel_name, I'm looking for something hotel_pricerange.	1	yes
Hotel	How many people are going to be staying?	It'll be hotel_people people. I'd like to stay at a hotel_stars star hotel.	2	no
Hotel	Would you like the hotel to have a specific number of stars?	We'd prefer if it had hotel_stars, however what's more important is that we're in the hotel_area part of the city.	1	no
Hotel	Which area would you like to stay in?	We'd prefer the hotel_area part of the city.	1	no
Hotel	When are you getting here?	We're getting here hotel_book_day and staying for hotel_stay days.	1	no
Train	How many people are travelling?	There's train_people of us, so we need to leave by train_leave_at.	1	no
Train	When do you need to arrive by?	We need to arrive by train_arrive_by.	1	no
Train	What day are you travelling?	We need to be there by train_arrive_by on train_day.	1	yes
Train	Where are you leaving from?	We're travelling on train_day. Leaving from train_departure.	2	yes
Train	Where do you want to go?	The train_people of us want to go to train_destination.	1	yes
Train	When do you want to leave?	I want to leave at train_leave_at in order to arrive by train_arrive_by.	1	no
Train	Did you need to arrive by a certain time?	Yeah we need to be there by train_arrive_by. There are train_people adults travelling.	2	no
Train	How many tickets do you want?	Not only do we need train_people tickets, but also we need to leave by train_leave_at.	1	no
Train	How many tickets do you need?	We need to go on train_day.	1	no
Train	Which day are you going?	We're going train_day, but we don't care when we leave as long as we get there by train_arrive_by.	1	no

Table 5: Single domain templates.

Domain	Question	Template	# Sentences	Reverse?
Restaurant-Hotel	How many people is the reservation for?	There's restaurant_people people, we'll also need a hotel room for the same number of people.	1	no
Restaurant-Hotel	What type of cuisine would you like to eat?	I'm in the mood for restaurant_food_types. I also need to book a hotel_pricerange hotel.	2	no
Restaurant-Hotel	When would you like to book your reservation for?	We need to check-in on hotel_book_day, let's book for restaurant_day.	1	yes
Restaurant-Hotel	What day would you like to book the reservation on?	I want to stay for hotel_stay days. I'd like the reservation on restaurant_day.	2	yes
Restaurant-Hotel	Do you have a specific restaurant in mind?	We'd like to stay in the hotel_area part of the city, so a reservation at restaurant_names would be nice.	1	yes
Restaurant-Hotel	How many people will be staying?	There's hotel_people of us, we need a dinner reservation restaurant_people as well.	1	no
Restaurant-Hotel	What day will you arrive?	hotel_book_day. I want to have dinner at restaurant_booking_time that day.	2	no
Restaurant-Hotel	Do you have a preference for the area the hotel is in?	Since we'd like to eat at restaurant_names I'd prefer the hotel_area area of the city.	1	yes
Restaurant-Hotel	How long are you planning on staying for?	hotel_stay days with dinner on restaurant_day.	1	no
Restaurant-Hotel	Do you have a specific hotel in mind?	I'd prefer hotel_name since it's close to my favourite restaurant restaurant_names.	1	no
Restaurant-Train	How many people is the reservation for?	train_people of us will be traveling there and restaurant_people of us will be eating.	1	no
Restaurant-Train	What type of cuisine would you like to eat?	I'm craving restaurant_food_types. We'll need train tickets for train_people people too.	2	no
Restaurant-Train	What time would you like to dine?	After we eat at restaurant_booking_time we need to book a train for train_leave_at.	1	no
Restaurant-Train	What type of cuisine do you like?	We're getting off the train at train-destination. A restaurant_food_types restaurant nearby would be nice.	2	yes
Restaurant-Train	When would you like to book your reservation?	restaurant_booking_time. We need a train arriving by train_arrive_by as well.	2	no
Restaurant-Train	How many people are travelling?	We want to have dinner on restaurant_day, there'll be train_people of us travelling there.	1	yes
Restaurant-Train	When do you need to arrive by?	To make our dinner reservation at restaurant_booking_time I want to arrive by train_arrive_by	1	yes
Restaurant-Train	What day are you travelling?	Our favourite restaurant, restaurant_names, is opening back up so let's go train_day.	1	yes
Restaurant-Train	Where are you leaving from?	Leaving from train_departure. We want to dine at restaurant_names.	2	no
Restaurant-Train	Where do you want to go?	I'm craving restaurant_food_types so let's head to train_departure.	1	yes
Hotel-Train	What is your pricerange?	Something hotel_pricerange. We also need a train book on train_day.	2	no
Hotel-Train	How many people going to be staying?	There's hotel_people. Can I also book a train to leave from train_departure?	2	no
Hotel-Train	Would you like the hotel to have a specific number of stars?	I need tickets for train_people people. We'd prefer hotel_stars stars.	2	yes
Hotel-Train	Which area would you like to stay in?	Our train is arriving by train_arrive_by, that means that the hotel_area part of the city would be best.	1	yes
Hotel-Train	When are you getting here?	Sometime on hotel_book_day, but we'll need a train for train_day.	1	no
Hotel-Train	When do you want to leave?	We want to stay for hotel_stay days and we need tickets for train_day.	1	yes
Hotel-Train	Did you need to arrive by a certain time?	We check-in on hotel_book_day. Let's arrive by train_arrive_by.	2	yes
Hotel-Train	Where are you going?	We'd like to go to train_destination despite being in the hotel_area part of the city.	1	no
Hotel-Train	How many tickets do you need?	train_people of us need to get to hotel_name.	1	no
Hotel-Train	Which day are you going?	Although we arrive hotel_book_day, I'd like to go train_day.	1	yes

Table 6: Dual domain templates.