Are LLMs all you need for TOD?

Vojtěch Hudeček & Ondřej Dušek

SIGDial 2023

Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

Problem introduction

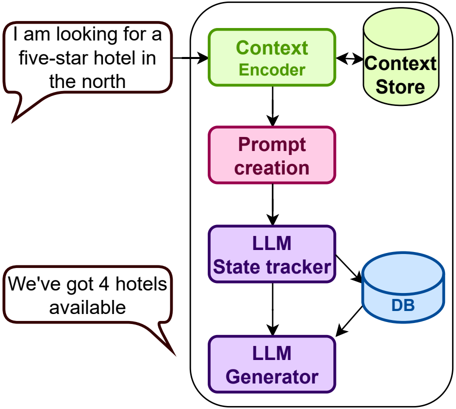
- Task-oriented dialogue
- Well-defined goal
- Explicit state, database access
- MultiWOZ 2.2, (Schema Guided Dataset)

User	I need to book a hotel in the east that has 4 stars.
System	I can help you with that. What is your price range?
State	restaurant {}, ..., hotel {"area": "east", "stars": "4" }
User	That doesn't matter as long as it has free wifi and parking.
System	If you'd like something cheap, I recommend the Allenbell.
State	restaurant {}, ..., hotel {"area": "east", "stars": "4", "wifi": yes", "parking": "yes"}
	...

Table 1: A simplified example taken from the MultiWOZ corpus

Architecture

- In-context learning
- Contextual retrieval of relevant examples
- Separate state decoding
- Context-aware response generation
- ChatGPT-0301, Tk-Instruct-11B, Alpaca-7B, GPT-NeoXT-20B



What this is NOT

- **We don’t perform LLM finetuning**
- We want to asses the model’s ability to deal with the task only via in-context learning
    - Potential universal usage

Retrieval

- A few examples are chosen to form a *context store (CS)*
  - CS divided by domains
  - domain detected by separate LLM call
- Conversation snippets are embedded and saved (with labels)
- retrieval of K most similar snippets at each step
- K-shot prompt

Prompt construction

- Task definition
- Domain description
- Conversation History
- User Utterance
- State & DB

Prompt	Definition: Capture values from a conversation about hotels. Capture pairs "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 2: 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.

Domain description

- List of pairs `slot:description`
  - domain ontology
- The descriptions capture slot semantics in natural language
- Domain selection performed by separate LLM call

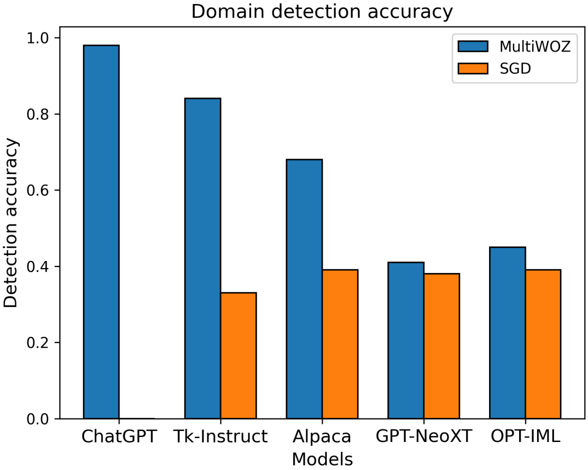


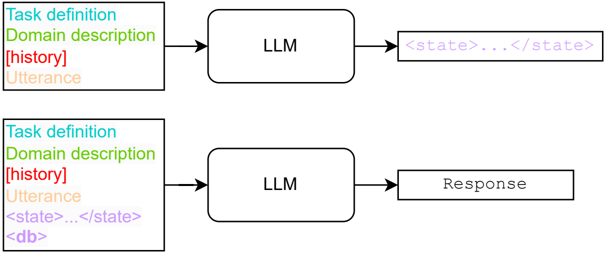
Figure 1: Domain detection accuracy

Database

- Well-structured list of entries with predefined keys
- The keys must be specified explicitly
- We perform approximate matching to account for spelling errors and minor mismatches (swimming pool vs. swimmingpool)
- Only count of matches included.

2-stage decoding

- State predicted separately
- Predicting only state updates



Results - zero shot

- ChatGPT-0301, Tk-Instruct-11B, Alpaca-7B

Model	oracle BS	Slot-F1	Success
Alpaca	✗	0.07	0.04
Tk-Instruct	✗	0.04	0.04
ChatGPT	✗	0.40	0.31
Alpaca	✓	–	0.08
Tk-Instruct	✓	–	0.18
ChatGPT	✓	–	0.47

Table 3: Results of selected models in a few shot setting

Results - few shot

- ChatGPT-0301, Tk-Instruct-11B, Alpaca-7B

Model	oracle BS	Slot-F1	Success
Alpaca	✗	0.08	0.06
Tk-Instruct	✗	0.33	0.19
ChatGPT	✗	0.51	0.44
Alpaca	✓	–	0.41
Tk-Instruct	✓	–	0.46
ChatGPT	✓	–	0.68

Table 4: Results of selected models in a few shot setting

Number of shots

- Introduction of examples helps
- Increased number of examples has a negligible contribution
- Differences consistent among models

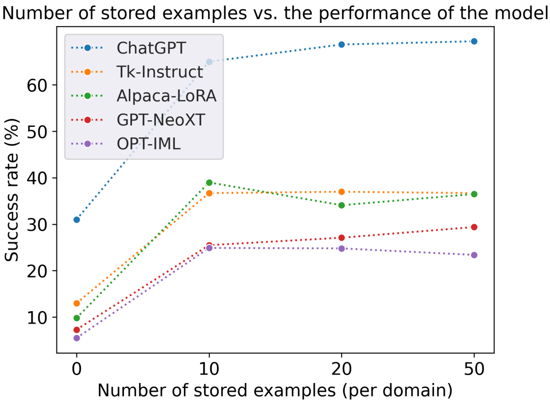


Figure 2: The influence of the number of available examples

Human Evaluation

- Humans evaluate success on sub-dialogue level
    - For each subdomain
- Humans try to clarify, rephrase
- ChatGPT performs best
- Better results than automated evaluation

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

Table 5: Results of human evaluation

Conclusion

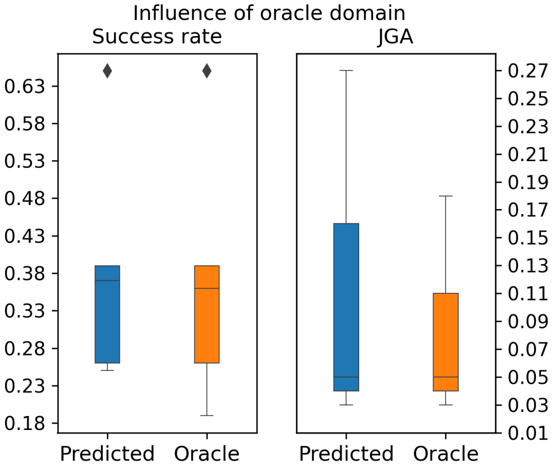
- Belief state tracking is very limited out of the box
- If provided with a correct belief state, the models can interact with the user successfully
- Examples in the prompt help, but the relevance is not super important
- In the interactive evaluation with human users the models performed better than assessed with the automatic metrics.
- Limitation: training data contamination

Thank you!

- The code available at `https://github.com/vojtsek/to-llm-bot`

Vojtěch Hudeček
hudecek@ufal.mff.cuni.cz
ufal.mff.cuni.cz/vojtech-hudecek

Using oracle domains



Response generation prompt

Prompt	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
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.