

# LLMs for Dialogue & Data-to-text

Ondřej Dušek

thanks: Vojtěch Hudeček, Damian Wałoszek, Ondřej Plátek, Karen Li, Simone Balloccu, Pier Balestrucci, Alessandro Mazzei, Luca Anselma, Jędrzej Warczyński, Mateusz Lango

NPFL140 LLMs

31/03/2025



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



unless otherwise stated

# Task-oriented Dialogue

- **Assistant:** fulfill user requests (book a hotel/restaurant/taxi etc.)
  - need to **connect to external DB & provide correct info**
  - MultiWOZ: benchmark for this, 10k dialogues

(Budzianowski et al., 2018)

<https://aclanthology.org/D18-1547/>

- user:** I am looking for a train from Cambridge to London Kings Cross.  
**state:** *{train {departure = cambridge, destination = london kings cross}}*  
**DB:** *{train (70) {...}}*

|                | [count]      | [departure]                                            | [destination]                      |
|----------------|--------------|--------------------------------------------------------|------------------------------------|
| <b>system:</b> | There are 70 | trains departing from Cambridge to London Kings Cross. | What day would you like to travel? |
- user:** I would like to leave on Saturday after 18:45.  
**state:** *{train {day = saturday, departure = cambridge, destination = london kings cross, leave at = 18:45}}*  
**DB:** *{train (3) {arrive by = 19:51,21:51,23:51; id = TR0427,TR0925,TR4898; leave at = 19:00,21:00,23:00; ... }}*

|                | [id]   | [leave_at]      | [arrive_by]                                               |
|----------------|--------|-----------------|-----------------------------------------------------------|
| <b>system:</b> | TR0427 | leaves at 19:00 | and arrives by 19:51 . Would you like to book this train? |
- user:** Yes, I would like to book it for eight people.

**system:** I have booked it for you. Your reference number is 00000057. Is there anything else I can help you with?
- user:** I am also looking for an expensive restaurant in the centre.  
**belief:** *{restaurant {area = centre, price range = expensive} train {...}}*  
**DB:** *{restaurant (33) {area = centre (33); name=Curry Garden, ...; ...}, ...}*

|                | [count]      | [price_range]                        | [area]                                             |
|----------------|--------------|--------------------------------------|----------------------------------------------------|
| <b>system:</b> | There are 33 | expensive restaurants in the centre. | Is there a particular type of food you would like? |

# Task-oriented Dialogue with LLMs

- How good are LLMs if we require structure?
  - slots / DB are given
  - no finetuning ~ **prompting only**
  - ChatGPT, Tk-Instruct, Alpaca... (7-20B params)
- A few examples in prompt (context store)
  - wide application potential
- Still the same idea: **context** → **state** → **DB** → **response**
  - additional step needed: domain detection

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:

instruction

domain

description

examples

dial. history

user input

- “pricerange”: the price of the hotel  
- “area”: the location of the hotel

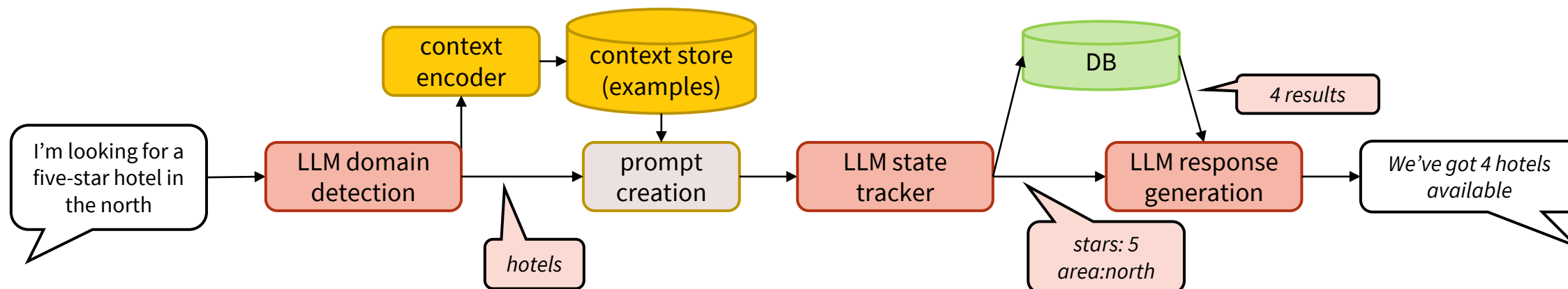
--- Example 1 ---

Assistant: “Hello, how can I help you?”

Customer: “I am looking for a five-star hotel in the north”

(Hudeček & Dušek, 2023)

<https://aclanthology.org/2023.sigdial-1.21>



# Results

- Domain detection ~ 70%+
  - Alpaca & TkInstruct OK, ChatGPT almost perfect
  - good enough to get relevant examples & prompts
- Belief state – not great
  - much worse than SotA
  - examples help (ChatGPT, TkInstruct: ~50-60% F1, Alpaca 8%), 10 ex./domain enough

- Responses:  
OKish

| Dialogue Success                          |                 | ChatGPT | TkInstruct |
|-------------------------------------------|-----------------|---------|------------|
| 1-step (corpus)                           | predicted state | 44%     | 19%        |
|                                           | gold state      | 68%     | 46%        |
| expert eval (end-to-end, with recoveries) |                 | 76%     | 64%        |

- More potential with better prompt engineering

# RAG-based open-domain dialogue

## Petbot – assistant for elderly

- in Czech
- prompted LLMs
  - Gemma, Aya – best Czech knowledge
- so far textual, audio hopefully coming
- 2-step:
  - detect rough intent
  - RAG (database of personal info)
  - use API (weather, time)
  - use (simulated) smart home sensors
  - direct responses (chitchat)

 **Petbot**

Odeslat

**Petbot:** Ahoj Ondřeji, vítěj! Jsem Petbot, tvůj přátelský asistent. Jak se dnes máš? 😊

**Ondřej:** Kolik je hodin?

**Petbot:** Je 00:13.

**Ondřej:** Kdy bude snídane?

**Petbot:** Snídane je v osm hodin.

**Ondřej:** Zavřel jsem okno?

**Petbot:** Okno je otevřené.

**Ondřej:** Jaké bude počasí?

**Petbot:** Dnes bude jasno a teplota bude 0 stupňů Celsia. Později se objeví déšť a teploty se budou pohybovat mezi -1 a 4 stupni Celsia.

(Waloszek et al., unpublished)

# LLMs for Rephrasing

- Simply ask LLMs to paraphrase stuff
- P-MultiWOZ: “generation-specific” responses for MultiWOZ
- Diet coaching chatbot: more variable responses

| User: Can you help me find a train going to Cambridge leaving on Saturday? |                                                                                                  |                                                                                         |
|----------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| System: What city will you be departing from?                              |                                                                                                  |                                                                                         |
| Generation                                                                 | Paraphrased Request                                                                              | Paraphrased Response                                                                    |
| Boomer                                                                     | Would it be possible for you to assist in locating a train departing for Cambridge on Saturday?  | In what city will the train departure for Cambridge on Saturday be originating from?    |
| Gen X                                                                      | Could you please assist in locating a train to Cambridge departing on Saturday?                  | What is the departure city for the train to Cambridge on Saturday?                      |
| Gen Y                                                                      | Hey there, can you assist in locating a train heading to Cambridge with a departure on Saturday? | What city are you departing from, for finding a train heading to Cambridge on Saturday? |
| Gen Z                                                                      | Hey there! Got any info on trains heading to Cambridge, departing on Saturday? 😊                 | What city are you departing from, for trains heading to Cambridge on Saturday? 😊        |

(Balestrucci et al., unpublished)

(Li, 2025)  
<https://dspace.cuni.cz/handle/20.500.11956/197096>

Update me on last week

rephrased

Response A

I'm digging into your food diary from 📅 May 13, 2024 to May 19, 2024 🗨️

📊 Some dates are missing from your data. If I fill in the gaps, it might not be exactly correct. Do you want me to make an educated guess for you? 🤖

templates

Response B

I'm analysing your data:

- from May 13 2024 to May 19 2024

and getting the insights... 🗨️

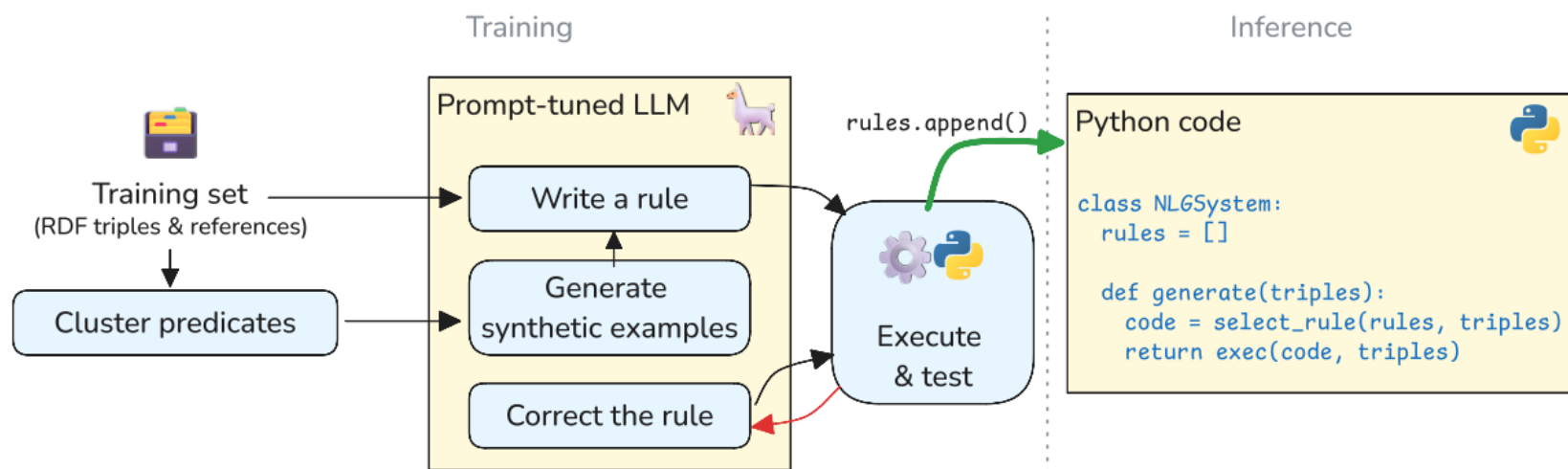
Some days are missing. I can estimate them but the final value could be different to reality. Would you like me to estimate your data?

# LLMs with code generation

- Generate Python rules to generate text

(Warczynski et al., 2024)

<https://aclanthology.org/2024.inlg-main.48>



(John Smith, award, Nobel Prize)  
(John Smith, alma mater, Oxford University)  
(John Smith, birth date, 25 October 1901)  
(John Smith, birth place, Stoke-on-Trent)

```
subj = triples[0][0]
birth_date = next(obj for subj, pred, obj in triples if pred == 'birth date')
birth_place = next(obj for subj, pred, obj in triples if pred == 'birth place')
alma_mater = next(obj for subj, pred, obj in triples if pred == 'alma mater')
award = next(obj for subj, pred, obj in triples if pred == 'award')

output = f"{subj}, born on {birth_date} in {birth_place}, graduated from {alma_mater}, his alma mater. He won the prestigious {award}."
```

(b) A rule for describing an input with the following set of relations: “alma mater”, “award”, “birth date” and “birth place”.