Combining Large Language Models and Structured Knowledge Representations for User-Personalized Conversations of In-Car Assistants

Johannes Kirmayr – MT Final Presentation

10.06.2024, Munich

Chair of Software Engineering for Business Information Systems (sebis)
Department of Computer Science
School of Computation, Information and Technology (CIT)
Technical University of Munich (TUM)
wwwmatthes.in.tum.de

Outline

- Motivation
- Research Questions
- Approach and Methodology
- Results and Discussion
- Conclusion and Future Work

Motivation

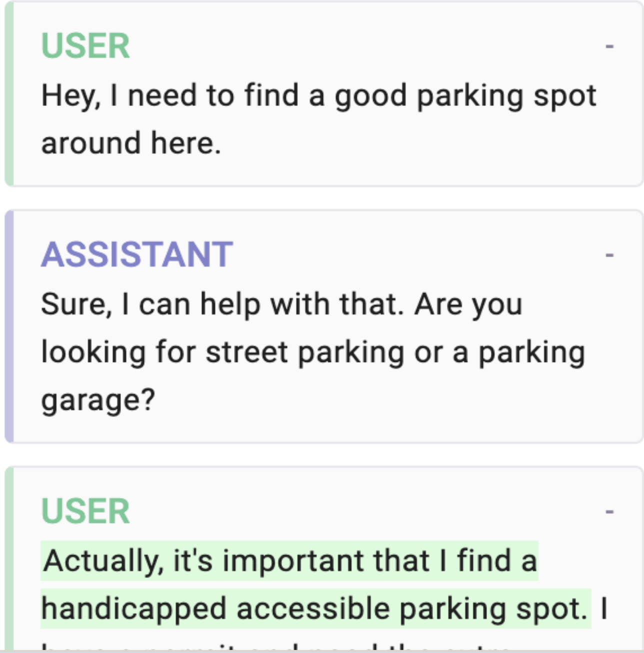
LLMs have revolutionized the generation and understanding of humanlike text

BMW is developing LLM-driven Intelligent Personal Assistant (IPA)

User Study

Status Quo LLM-IPA-Prototype

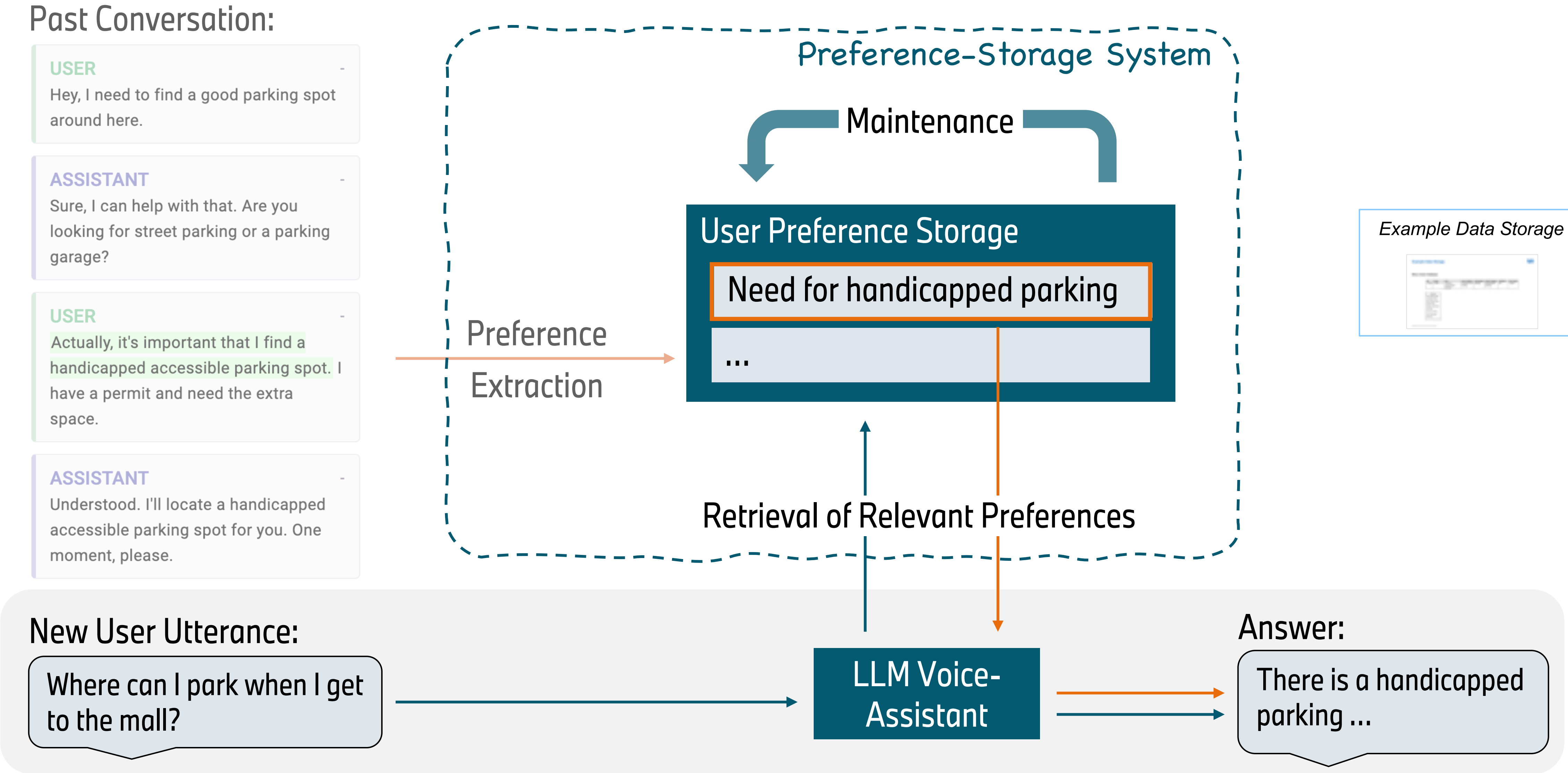
- Conversation-Only Memory
  - Memory is not preserved after a conversation restart
- Lack of Personalized Interactions
  - No recall of user preferences for long-term relationship



Thesis:

- Car-domain dataset to evaluate long-term preference memory
- Preference-Storage system for personalization

Motivation – Preference-Storage System



Example Data Storage

Outline

- Motivation
- Research Questions
- Approach and Methodology
- Results and Discussion
- Conclusion and Future Work

Research Questions

RQ1 How can a suitable conversational dataset be created
to develop and evaluate the personal preference memory system?

RQ2 How could personal preferences be effectively extracted from conversations and stored?

RQ3 Which methods enable a context-related retrieval of preferences?

RQ4 Which method can be used to effectively maintain the preference storage
when personal preferences change over time?

Outline

- Motivation
- Research Questions
- Approach and Methodology
- Results and Discussion
- Conclusion and Future Work

RQ1: Dataset

How can a suitable conversational dataset be created to develop and evaluate the personal preference memory system?

Goal	Generate synthetic conversational dataset **grounded** on BMW data insights

Requirements

- User-Assistant dialogues containing user preferences
- Conversations realistic for the in-car setting (closed-domain)
- Interaction over different time scales
  - to test extraction, later retrieval, and change of preferences (maintenance)

Why synthetic?

- Real system does not exist
- Related work shows synthetic conversational dataset can outperform crowdsourced [1]
- Cost, Time

[1] X. Xu, Z. Gou, W. Wu, Z.-Y. Niu, H. Wu, H. Wang, and S. Wang. “Long Time No See! Open-Domain Conversation with Long-Term Persona Memory”. In: Findings of the Association for Computational Linguistics: ACL 2022. Ed. by S. Muresan, P. Nakov, and A. Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 2639–2650. doi: 10 . 18653 / v1 / 2022 . findings - acl . 207. url: https : / / aclanthology.org/2022.findings-acl.207.

RQ1: Resulting Dataset Parts

How can a suitable conversational dataset be created to develop and evaluate the personal preference memory system?

Targeted evaluation

1 *Extraction* In-Car Conversation

A) *One-Preference*

USER -
Hey, I need to find a good parking spot around here.

ASSISTANT -
Sure, I can help with that. Are you looking for street parking or a parking garage?

USER -
Actually, it's important that I find a handicapped accessible parking spot. I have a permit and need the extra space.

ASSISTANT -
Understood. I'll locate a handicapped accessible parking spot for you. One moment, please.

B) *No-Preference*

USER -
What's the latest in music

ASSISTANT -
The latest trend in music includes a mix of new album releases and emerging artists. Would you like to listen to some new tracks?

USER -
Current top charts

ASSISTANT -
I can play the current top chart hits for you. Starting playback now.

2 *Retrieval* Utterance (New Timescale)

USER: Can you find good parking nearby?

3 *Maintenance* Utterances (New Timescale)

Equal Pref.: USER: Can you find handicapped parking nearby?

Negate Pref.: USER: Can you find regular parking, handicapped not needed?

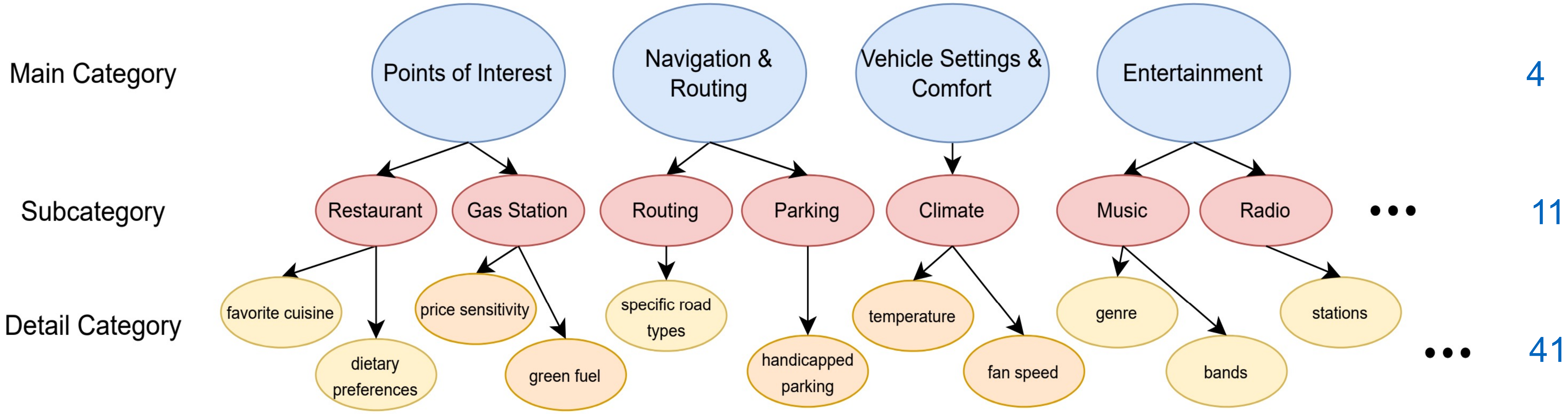
Different Pref.: USER: Can you always find off-street parking for me?

RQ1: In-Car Preference Categories

How can a suitable conversational dataset be created to develop and evaluate the personal preference memory system?

Manually defined categories relevant for in-car assistant
- together with BMW IPA Expert
- based on most used in-car intents (bottom-up)

closed-domain



Full List of Categories

RQ1: Dataset Generation

How can a suitable conversational dataset be created to develop and evaluate the personal preference memory system?

User Profile			
User Profile Data			
Age	40		
Technological Proficiency	High		
Conversation Style	Commanding		
Location	Tokyo, Japan		
Preferences (10)			
Main Category	Subcategory	Detail Category	Attribute
Points of Interest	Restaurant	Favorite Cuisine	Italian
Points of Interest	Restaurant	Favorite Cuisine	Mexican
⋮	⋮	⋮	⋮
Navigation & Routing	Parking	Handicapped parking	Yes

Consistency per user
Diversity across users

user profile

user preference

conversation criteria

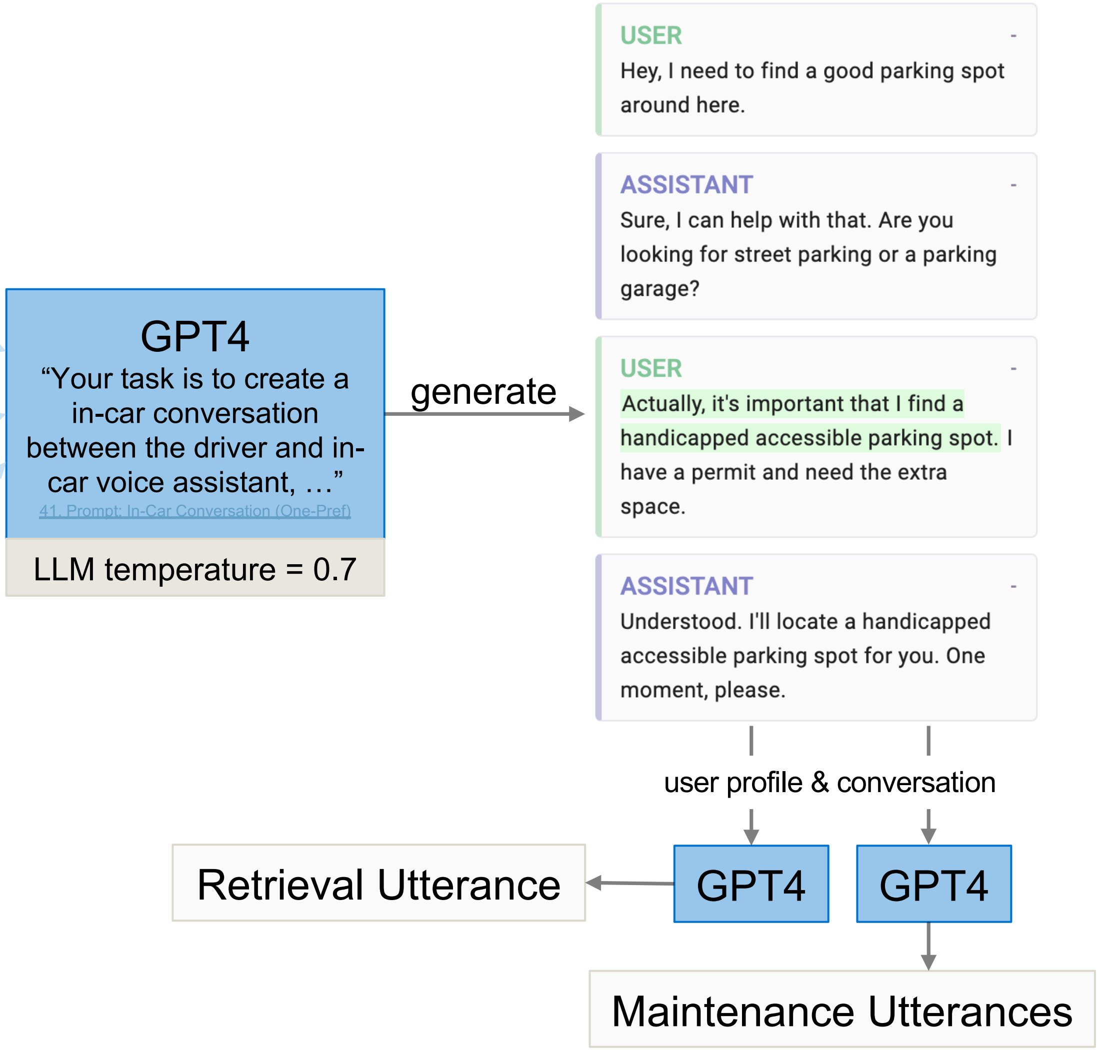
Conversation Criteria

Conversation Length: 4 turns
Position User Preference: 3rd turn
…
Few Shot Example: …
(real conversation turn)

Diversity across conversations

Realism

34. Sampling of Dynamic Inputs Per Conversation

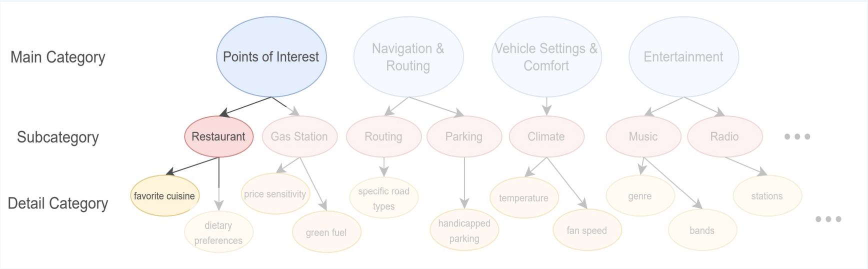


RQ2: Preference Extraction + User Preference Storage

How could personal preferences be effectively extracted from conversations and stored?

Extraction – OpenAI function calling

Closed-Domain: only extract in predefined categories



Function Definition

```
"type": "function",
"function": {
    "name": "extract_user_preference",
    "description": "A function that extracts personal preferences
        of the user in the categories 'Points of Interest', '
        Navigation and Routing', 'Vehicle Settings and Comfort', '
        Entertainment and Media'.",
    "parameters": <our defined pydantic schema>
```

Parameter Schema

convert to
OpenAI function schema

```
class OutputFormat(BaseModel):
    user_sentence_preference_revealed: Optional[str] = Field(default=None,
        description="user sentence where the user revealed the preference.")
    user_preference: Optional[str] = Field(default=None,
        description="The preference of the user.")

class Restaurant(BaseModel):
    favourite_cuisine: Optional[List[OutputFormat]] = Field(default=[],
        description="The user's preference in the topic 'Favourite Cuisine'.",
        examples=EXAMPLES["favourite_cuisine"])

class PointsOfInterest(BaseModel):
    restaurant: Optional[Restaurant] = Field(default=None,
        description="The user's preferences in the category 'Restaurant'. This includes
            preferences in the topics 'Favourite Cuisine', 'Preferred Restaurant Type', [...]."
            )

class PreferencesFunctionOutput(BaseModel):
    points_of_interest: Optional[PointsOfInterest] = Field(default=None,
        title = "Preferences Points of Interest",
        description="The user's preferences in the category 'Points of Interest'. This includes
            preferences in the topics 'Restaurant', 'Gas Station', [...].",)
```

In-Car Conversation:
USER: I've been craving Italian food, can . . .
ASSISTANT: Sure, . . .

→

GPT4
with function
calling
47. Prompt: Extraction
temperature = 0.0

generate →

```
{
  "points_of_interest": {
    "restaurant": {
      "favourite_cuisine": [
        {
          "user_sentence_preference_revealed": "I've been craving italian food",
          "user_preference": "Italian"
        }]}}}
```

RQ2&3: User Preference Storage + Preference Retrieval

How could personal preferences be effectively extracted from conversations and stored?
Which methods enable a context-related retrieval of preferences?

Storage

pk	vector	text	main category	subcategory	detail category	attribute	user name
<uuid>	[-0.006..., ...]	I've been craving italian food	points of interest	restaurant	favourite cuisine	Italian	<uuid>
<uuid>	[-0.001..., ...]	I just want the temperature to stay at 21 degrees ...	vehicle settings and comfort	climate control	preferred temperature	21 degrees	<uuid>

Search
cosine similarity

Query
main- and subcategory

A) Retrieval by embedding

[-0.005…, …]

Retrieval Utterance

Find me a restaurant close by

B) Retrieval by function-calling

OpenAI Function Calling

```
{
  "points_of_interest": {
    "restaurant": True
  }
}
```

[1] W. Zhong, L. Guo, Q. Gao, H. Ye, and Y. Wang. "MemoryBank: Enhancing Large Language Models with Long-Term Memory". In: Proceedings of the AAAI Conference on Artificial Intelligence 38.17 (Mar. 2024), pp. 19724–19731. doi: 10.1609/aaai.v38i17. 29946. url: https://ojs.aaai.org/index.php/AAAI/article/view/29946.
[2] B. Wang, X. Liang, J. Yang, H. Huang, S. Wu, P. Wu, L. Lu, Z. Ma, and Z. Li. Enhancing Large Language Model with Self-Controlled Memory Framework. 2024. arXiv: 2304.13343 [cs.CL].

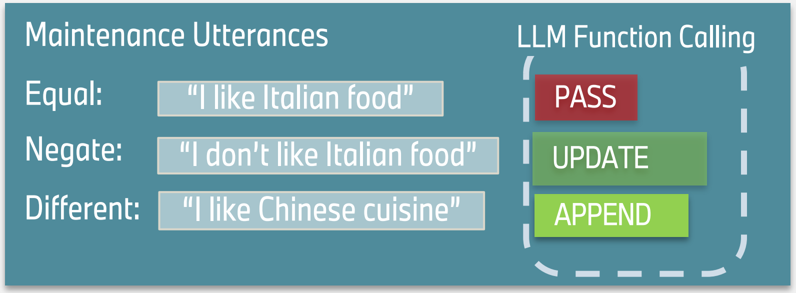
RQ4: User Preference Storage Maintenance

Which method can be used to effectively maintain the preference storage when personal preferences change over time?

Maintenance [1]

Motivation: Ever-growing storage with redundant or even contradictory preferences

Existing Preference: "I like Italian cuisine"



Detailed Maintenance

[1] Bae, S., Kwak, D., Kang, S., Lee, M. Y., Kim, S., Jeong, Y., Kim, H., Lee, S.-W., Park, W., & Sung, N. (2022). *Keep Me Updated! Memory Management in Long-term Conversations*. http://arxiv.org/abs/2210.08750

Outline

- Motivation
- Research Questions
- Approach and Methodology
- Results and Discussion
- Conclusion and Future Work

RQ1: Dataset

How can a suitable conversational dataset be created to develop and evaluate the personal preference memory system?

Dataset Statistics:

- Dataset consists of

1000	*One-Prefence* Conv.
1000	Retrieval Utterance
1000 (x3)	Maintenance Utterances
100	*No-Preference* Conv.

- Total cost of generation is ~23$ (Azure GPT-4-Turbo)

Human Evaluation:

- 3 Judges (BMW IPA Experts)
- Systematic evaluation on subset of 45 data points
- Score: Likert scale 1 (worst) – 3 (best); or binary

Results:

- *In-Car Conversations* were scored highly realistic (2.83 / 3)
- User preference was mostly clear and natural (2.57 / 3)
- 6 x preference was not clear enough, 3 x multiple preferences revealed

- *Retrieval utterance* was scored valid for dataset in 95%
- *Maintenance utterances* were scored valid for dataset in 100%

Evaluation Interface

Detailed Dataset Evaluation

Qualitative Analysis

RQ2: Preference Extraction – Quantitative Evaluation

How could personal preferences be effectively extracted from conversations and stored?

Setting Evaluation on 400 *In-Car Conversations* with preference and 50 without preference

In-Car Conversation with user preference

Category level	#categories	Precision		Recall	
		gpt-4-turbo	gpt-3.5-turbo	gpt-4-t.	gpt-3.5-t.
Main (All)	4	**0.94**	0.94	**0.98**	0.79
Sub (All)	11	**0.89**	0.88	**0.97**	0.76
Detail (All)	41	**0.75**	0.64	**0.91**	0.64

In-Car Conversation without user preference

- Risk of extraction if no preference is present is high (**32%**)

Further Findings (gpt-4-t.)

- Generated JSON output adheres in 97% to the parameter schema
- Median tokens per extraction:
  - 8278 (7938 come from parameter schema)
  - Cost: 0.08$ per extraction

Detailed Evaluation

Qualitative Evaluation

RQ3: Preference Retrieval

Which methods enable a context-related retrieval of preferences?

Setting

Evaluation on 289 *Retrieval Utterances*
On average 7.3 preferences per user stored

Vector Embedding-Based Retrieval

- Latency: 0.26s
- Top-k accuracy, k=#ground truth subcat. pref.:
  - Only text: 0.77
  - Detail category + attribute + text: **0.86**
  - Why is it 0.09 better?

EB-T Embedding String: "I always find NavFlow to be reliable."

EB-DAT Embedding String: "traffic_information_source_preferences: NavFlow. I always find NavFlow to be reliable."

- Disambiguation
- Richer Semantic Representation
- Consistency in Retrieval

LLM Function-Call-Based Retrieval

- Latency: 2.08s
- Precision: 0.82
- Recall: 0.94

Detailed Evaluation

Qualitative Evaluation

RQ4: User Preference Storage Maintenance

Which method can be used to effectively maintain the preference storage when personal preferences change over time?

Setting	Evaluation on 829 *Maintenance Utterances*

Comparison with maintenance vs. without:

- 89% less redundant preferences
- 86% less contradictory preferences
- 83% less preferences in total

Detailed Evaluation

Outline

- Motivation
- Research Questions
- Approach and Methodology
- Results and Discussion
- Conclusion and Future Work

Conclusion and Future Work

Main Contributions

Synthetic Generation of Dataset
Targeted evaluation of basic preference-memory components

Baseline Benchmark on Dataset
Justifying utility of dataset
Using and evaluating state-of-the-art LLM methods

Benefit of Structured Preference Representation
Structured extraction method allows for transparency, controllability
Use of structure for improvement in retrieval and maintenance

Future Work

Improve Preference-Storage System
Over-extraction of preferences, increase precision

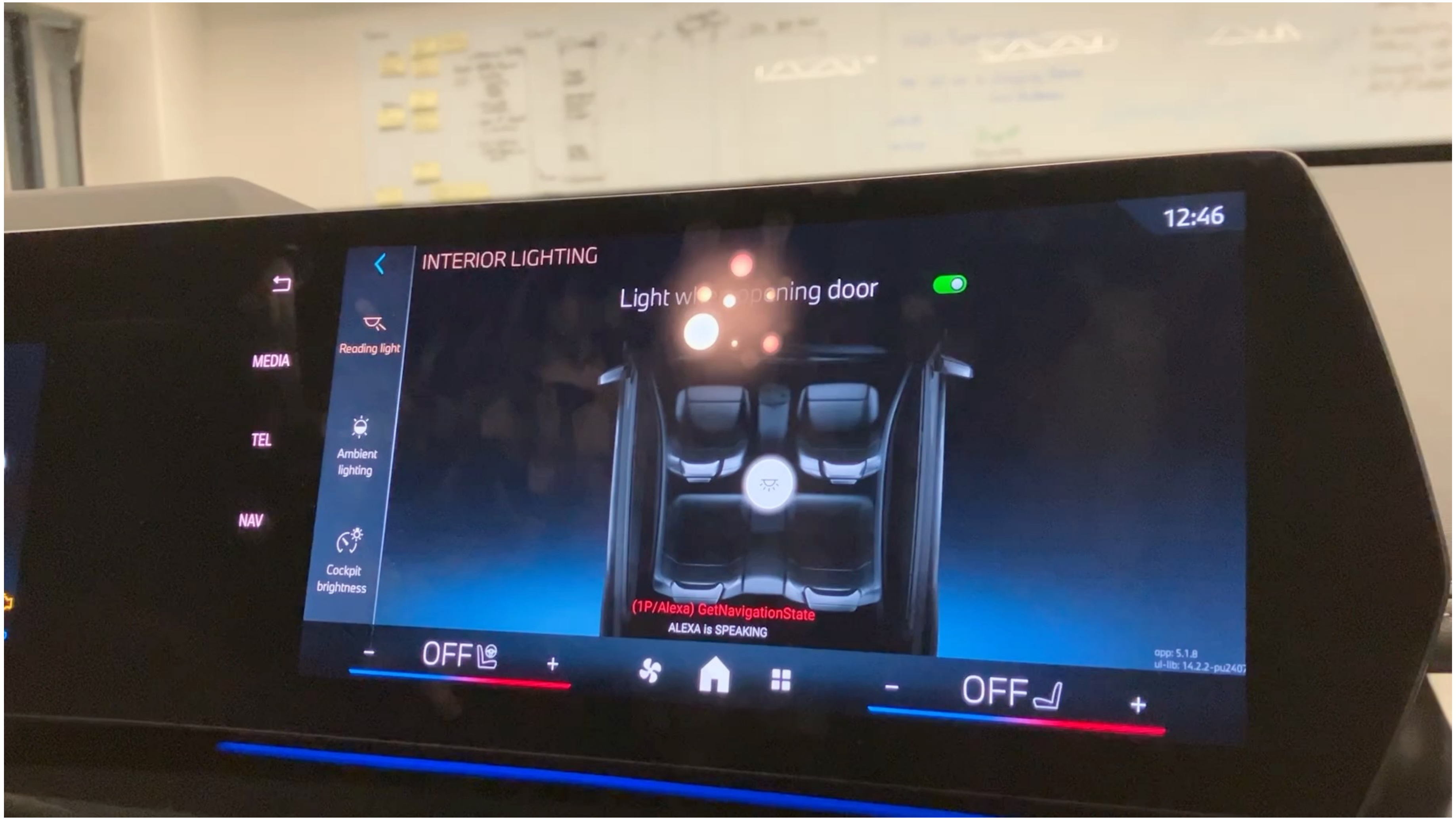
Evaluate where LLM necessary
Fine-tune smaller language model (SLM)
Replace with special-purpose model

Operationalize preference-storage system
How to use preference to increase user experience

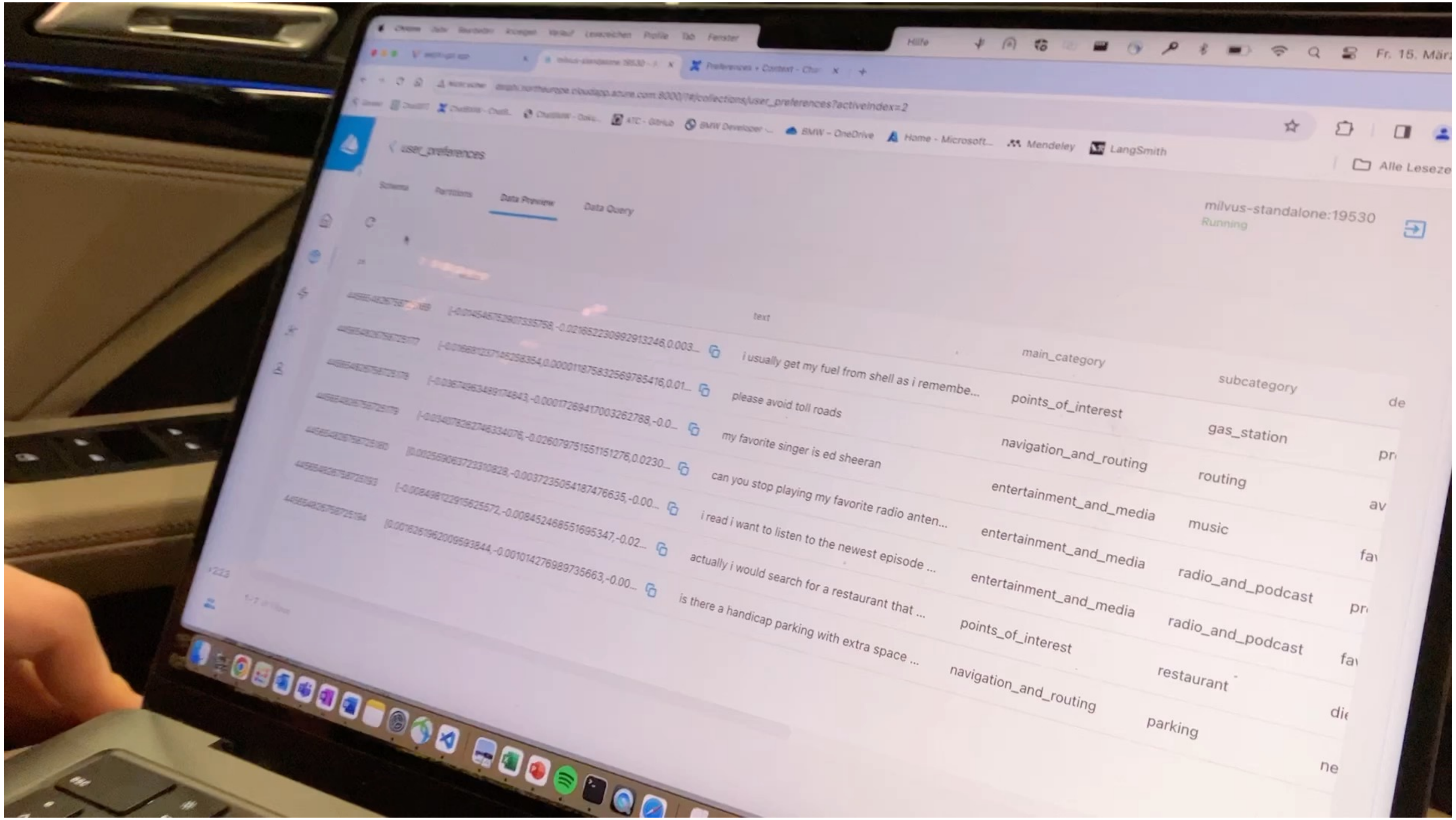
[1] Y. Chen, Q. Fu, Y. Yuan, Z. Wen, G. Fan, D. Liu, D. Zhang, Z. Li, and Y. Xiao. “Hallucination detection: Robustly discerning reliable answers in large language models”. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. 2023, pp. 245–255.
[2] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. “Chain-of-thought prompting elicits reasoning in large language models”. In: Advances in neural information processing systems 35 (2022), pp. 24824–24837.
[3] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models. 2023. arXiv: 2203.11171 [cs.CL].

DEMO

DEMO


12:46
INTERIOR LIGHTING
MEDIA
TEL
NAV
Reading light
Ambient lighting
Cockpit brightness
(1P/Alexa) GetNavigationState
ALEXA is SPEAKING
OFF
OFF

DEMO



user_preferences
Data Preview
Data Query
milvus-standalone:19530
Running
BMW - OneDrive
Mendeley
LangSmith
text
main_category
subcategory
i usually get my fuel from shell as i remembe...
points_of_interest
gas_station
please avoid toll roads
navigation_and_routing
routing
my favorite singer is ed sheeran
entertainment_and_media
music
can you stop playing my favorite radio anten...
entertainment_and_media
radio_and_podcast
i read i want to listen to the newest episode ...
entertainment_and_media
radio_and_podcast
actually i would search for a restaurant that ...
points_of_interest
restaurant
is there a handicap parking with extra space ...
navigation_and_routing
parking

DEMO

INTERIOR LIGHTING
12:57
MEDIA
TEL
NAV
Reading light
Ambient lighting
Cockpit brightness
(1P/Alexa) GetNavigationState
ALEXA is SPEAKING
OFF
OFF
MAX

Prof. Dr.

Florian Matthes

Technical University of Munich (TUM)
TUM School of CIT
Department of Computer Science (CS)
Chair of Software Engineering for Business
Information Systems (sebis)

Boltzmannstraße 3
85748 Garching bei München

+49.89.289.17132
matthes@in.tum.de
wwwmatthes.in.tum.de

Appendix

Literature Research

Lead Paper:

- Dataset: Target-oriented Proactive Dialogue Systems with Personalization: Problem Formulation and Dataset Curation [1]
- Preference Extraction: MemGPT: Towards LLMs as Operating Systems. [2]
- Preference Retrieval: Generative Agents: Interactive Simulacra of Human Behavior [3]
- Preference Storage Maintenance: Keep Me Updated! Memory Management in Long-term Conversations [4]

Complementary Literature:

- Datensatz: Personalizing Dialogue Agents: I have a dog, do you have pets too? [5]
- Präferenz Retrieval: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks [6]
- Präferenz Extraktion: Getting To Know You: User Attribute Extraction from Dialogues [7]

Contribution Masterthesis:

- Closed World: predefined categories for dataset and extraction with focus on the automotive domain
- Dataset of in-car conversations containing user preferences
- Focus on LLM approaches for components of the preference system

[1] Wang, J., Cheng, Y., Lin, D., Leong, C., & Li, W. (2023). Target-oriented Proactive Dialogue Systems with Personalization: Problem Formulation and Dataset Curation. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 1132–1143, Singapore. Association for Computational Linguistics.
[2] Packer, C., Fang, V., Patil, S. G., Lin, K., Wooders, S., & Gonzalez, J. E. (2023). MemGPT: Towards LLMs as Operating Systems. http://arxiv.org/abs/2310.08560
[3] Park, J. S., O'Brien, J., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative Agents: Interactive Simulacra of Human Behavior. Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, 1–22. https://doi.org/10.1145/3586183.3606763
[4] Bae, S., Kwak, D., Kang, S., Lee, M., Kim, S., Jeong, Y., Kim, H., Lee, S., Park, W., & Sung, N. (2022). Keep Me Updated! Memory Management in Long-term Conversations. In Findings of the Association for Computational Linguistics: EMNLP 2022, pages 3769–3787, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
[5] Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., & Weston, J. (2018). Personalizing Dialogue Agents: I have a dog, do you have pets too?. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.
[6] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-T., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf
[7] Wu, C., Madotto, A., Lin, Z., Xu, P., & Fung, P.. 2020. Getting To Know You: User Attribute Extraction from Dialogues. In Proceedings of the Twelfth Language Resources and Evaluation Conference, pages 581–589, Marseille, France. European Language Resources Association.

Approach & Methodology

Example Data Storage

Milvus Vector Database:

pk	vector	text	main category	subcategory	detail category	attribute	user name
<uuid>	[-0.006..., ...]	Need for handicapped parking	navigation and routing	parking	handicapped parking	Yes	<uuid>

Preference
+ main_category: String + subcategory: String + detail_category: String + attribute: String + category_type: Boolean + text: String + primary_key: String + user_name: String + vector: List

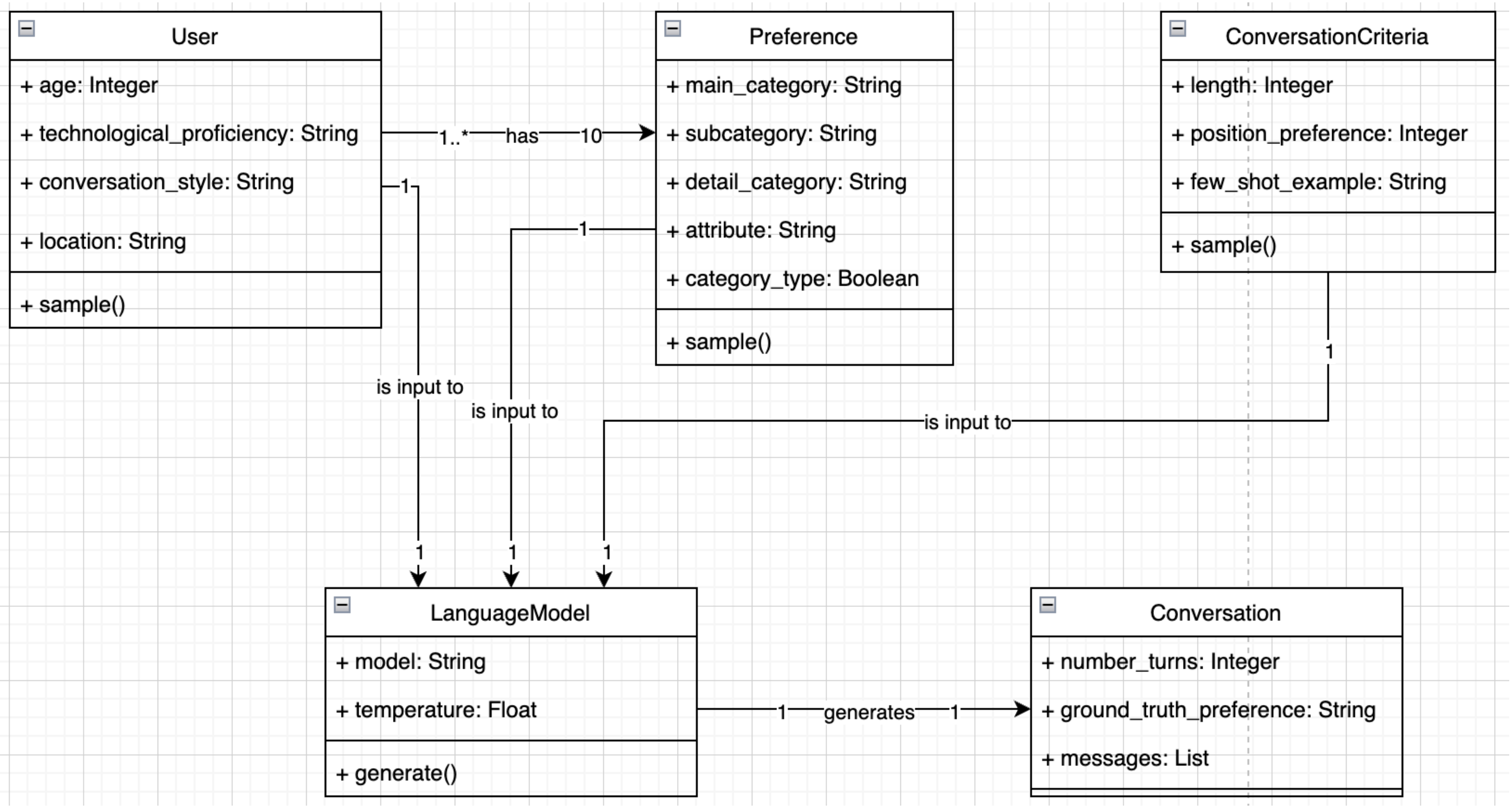
Dataset – Full list of Categories

1. Points of Interest:
   1. Restaurant:
      1. MP: Favorite Cuisine: Italian, Chinese, Mexican, Indian, American
      2. MP: Preferred Restaurant Type: Fast food, Casual dining, Fine dining, Buffet
      3. MP: Fast Food Preference: BiteBox Burgers, GrillGusto, SnackSprint, ZippyZest, WrapRapid
      4. MNP: Desired Price Range: cheap, normal, expensive
      5. MP: Dietary Preferences: Vegetarian, Vegan, Gluten-Free, Dairy-Free, Halal, Kosher, Nut Allergies, Seafood Allergies
      6. MNP: Preferred Payment method: Cash, Card
   2. Gas Station:
      1. MP: Preferred Gas Station: PetroLux, FuelNexa, GasGlo, ZephyrFuel, AeroPump
      2. MNP: Willingness to Pay Extra for Green Fuel: Yes, No (cheapest prefered)
      3. MNP: Price Sensitivity for Fuel: Always cheapest, Rather cheapest, Price is irrelevant
   3. Charging Station (in public):
      1. MP: Preferred Charging Network: ChargeSwift, EcoPulse Energy, VoltRise Charging, AmpFlow Solutions, ZapGrid Power
      2. MNP: Preferred type of Charging while traveling: AC, DC, HPC
      3. MNP: Preferred type of Charging when being at everyday points (f.e. work, grocery, restaurant): AC, DC, HPC
      4. MP: Charging Station Amenities: On-site amenities (Restaurant/cafes), Wi-Fi availability, Seating area, Restroom facilities
   4. Grocery Shopping:
      1. MP: Preferred Supermarket Chains: MarketMingle, FreshFare Hub, GreenGroove Stores, BasketBounty Markets, PantryPulse Retail
      2. MNP: Preference for Local Markets/Farms or Supermarket: Local Markets/Farms, Supermarket
2. Navigation and Routing:
   1. Routing:
      1. MP: Avoidance of Specific Road Types: Highways, Toll roads, Unpaved roads
      2. MNP: Priority for Shortest Time or Shortest Distance: Shortest Time, Shortest Distance
      3. MNP: Tolerance for Traffic: Low, Medium, High
   2. Traffic and Conditions:
      1. MNP: Traffic Information Source Preferences: In-car system, NavFlow Updates, RouteWatch Alerts, TrafficTrendz Insights
      2. MNP: Willingness to Take Longer Route to Avoid Traffic: Yes, No (traffic tolerated for fastest route)
   3. Parking:
      1. MNP: Preferred Parking Type: On-street, Off-street, Parking-house
      2. MNP: Price Sensitivity for Paid Parking: Always considers price first, Sometimes considers price, Never considers price
      3. MNP: Distance Willing to Walk from Parking to Destination: less than 5 min (accepting possible higher cost), less than 10 min (accepting possible higher cost), not relevant (closest with low cost)
      4. MNP: Preference for Covered Parking: Yes, No (doesn’t matter)
      5. MNP: Need for Handicapped Accessible Parking: Yes
      6. MNP: Preference for Parking with Security: Yes, No (doesn't matter)
3. Vehicle Settings and Comfort:
   1. Climate Control:
      1. MNP: Preferred Temperature: 18 degree Celsius, 19 degree Celcius, 20 degree Celcius, 21 degree Celcius, 22 degree Celcius, 23 degree Celcius, 24 degree Celcius, 25 degree Celcius
      2. MNP: Fan Speed Preferences: Low, Medium, High
      3. MNP: Airflow Direction Preferences: Face, Feet, Centric, Combined
      4. MNP: Seat Heating Preferences: Low, Medium, High
   2. Lighting and Ambience:
      1. MNP: Interior Lighting Brightness Preferences: Low, Medium, High
      2. MNP: Interior Lighting Ambient Preferences: Warm, Cool,
      3. MP: Interior Lightning Color Preferences: Red, Blue, Green, Yellow, White, Pink
4. Entertainment and Media:
   1. Music:
      1. MP: Favorite Genres: Pop, Rock, Jazz, Classical, Country, Rap
      2. MP: Favorite Artists/Bands: Max Jettison (Pop), Melody Raven (Pop), Melvin Dunes (Jazz), Ludwig van Beatgroove (Classical), Wolfgang Amadeus Harmonix (Classical), Taylor Winds (Country/Pop), Ed Sherwood (Pop/Folk), TwoPacks (Rap)
      3. MP: Favorite Songs: Envision by Jon Lemon (Rock), Dreamer's Canvas by Lenny Visionary (Folk), Jenny's Dance by Max Rythmo (Disco), Clasp My Soul by The Harmonic Five (Soul), Echoes of the Heart by Adeena (R&B), Asphalt Anthems by Gritty Lyricist (Rap), Cosmic Verses by Nebula Rhymes (Hip-Hop/Rap)
      4. MNP: Preferred Music Streaming Service: SonicStream, MelodyMingle, TuneTorrent, HarmonyHive, RhythmRipple
   2. Radio and Podcasts:
      1. MNP: Preferred Radio Station: EchoWave FM, RhythmRise Radio, SonicSphere 101.5, VibeVault 88.3, HarmonyHaven 94.7
      2. MP: Favorite Podcast Genres: News, Technology, Entertainment, Health, Science
      3. MP: Favorite Podcast Shows: GlobalGlimpse News, ComedyCraze, ScienceSync, FantasyFrontier, WellnessWave
      4. MNP: General News Source: NewsNexus, WorldPulse, CurrentConnect, ReportRealm, InfoInsight

Example Intents

Nav_SetDestinationByServer
Com_Phone_DialByContact
Nav_SetDestinationByHomeAddress
Nav_SetDestinationByServerSlot
Ent_Radio_SetStationbyName
Nav_RouteGuidanceOff
Gen_Cancel
Gen_ThirdParty_Keyword
Gen_Confirm
Gen_SelectEntryByLineNumberMediumSlot
Gen_MixContent
Ent_PlayMusic
Nav_SetDestinationByOfficeAddress

UML Class Diagram Dataset Generation



Sampling of User Profile Data

User Data:

- **Age:** Randomly sampled from a range of 20 to 90 years, in increments of 10.
- **Technological Proficiency:** Randomly sampled from three levels of technological proficiency: low, middle, and high.
- **Conversation Style:** The user conversation styles is randomly sampled from the following options:
  - *Keyword only:* Direct and to-the-point communication.
  - *Commanding:* Straightforward, imperative sentences.
  - *Questioning:* Seeking information or clarification.
  - *Conversational:* Casual, human-like manner.

  This represents the usual interaction style when dealing with current intent-based voice assistants (keyword only/commanding) to the enabled interaction possibility with LLM-based voice assistants (questioning/conversational).
- **Location:** Randomly sampled location from world-cities database [50].

Preferences:

Main Category	Subcategory	Detail Category	Category Type	Attribute	Weight
Points of Interest	Restaurant	Favorite Cuisine	**MP**	Italian	0.005
Points of Interest	Restaurant	Favorite Cuisine	**MP**	Chinese	0.005
Points of Interest	Restaurant	Favorite Cuisine	**MP**	Mexican	0.005
Points of Interest	Restaurant	Favorite Cuisine	**MP**	Indian	0.005
Points of Interest	Restaurant	Favorite Cuisine	**MP**	American	0.005
		⋮			
Navigation & Routing	Parking	Handicapped parking	**MNP**	Yes; No	0.024
		⋮			

$$w(r) = \frac{1}{\text{total number of detail categories}} * \frac{1}{\text{number of rows of current detail category}(r)}$$

Keyword

{"USER": "Look for ZephyrFuel stations"},
{"ASSISTANT": "I found a ZephyrFuel station 8 kilometers from here with good prices. Shall I set the course?"}

Commanding

{"USER": "Find me a kosher restaurant around here."},
{"ASSISTANT": "The closest kosher restaurant to our current location in Godda, India is Star Kosher Bites, which is approximately 20 minutes away by car. Would you like directions to this restaurant?"}

Questioning

{"USER": "I'm thinking about picking up some groceries too. What are my options?"},
{"ASSISTANT": "There are various grocery stores in the area such as FreshFarm Groceries, QuickMart, and GreenLeaf Store."}

Conversational

{"USER": "Hey, I just got this car and I'm still figuring things out. Can you help me adjust some settings?"},
{"ASSISTANT": "Of course, I'm here to assist you. Which settings would you like to adjust?"}

Sampling of Dynamic Inputs Per Conversation

- **Conversation Length:** Specifies the number of messages the conversation should have. Randomly sampled from a range of 2 to 8 messages, in increments of 2.
- **Position User Preference:** Specifies in which user message the preference should be revealed. Randomly sampled from a range of 2 to *Conversation Length*, with increment of 2.
- **Preference Strength Modulation:** Specifies how much the preference should be emphasized. Randomly sampled from: subtly hinted at, clearly stated, strongly emphasized.
- **Level of Proactivity Assistant:** Specifies how proactive the assistant acts. Randomly sampled from: medium, high, very high - no questions.
- **Few Shot Example:** Based on the subcategory of the preference, one few-shot example is randomly sampled. The list of few-shot examples is created from real in-car conversations and can be seen in section B.2.

Example Few-Shot Examples:

- **Restaurant**
  1. **User:** Show me restaurants nearby; **Assistant:** I have found multiple destinations in the category Restaurants. Which one should I select?
  2. **User:** I'm really hungry right now; **Assistant:** I have found these destinations for the category restaurant. Which one should I select?
  3. **User:** Fast Food Breakfast; **Assistant:** I have only found the destination Razz's Breakfast Bar and Grill, 5307, Westfair Ave, Schofield, Wisconsin. Should I start the guidance?

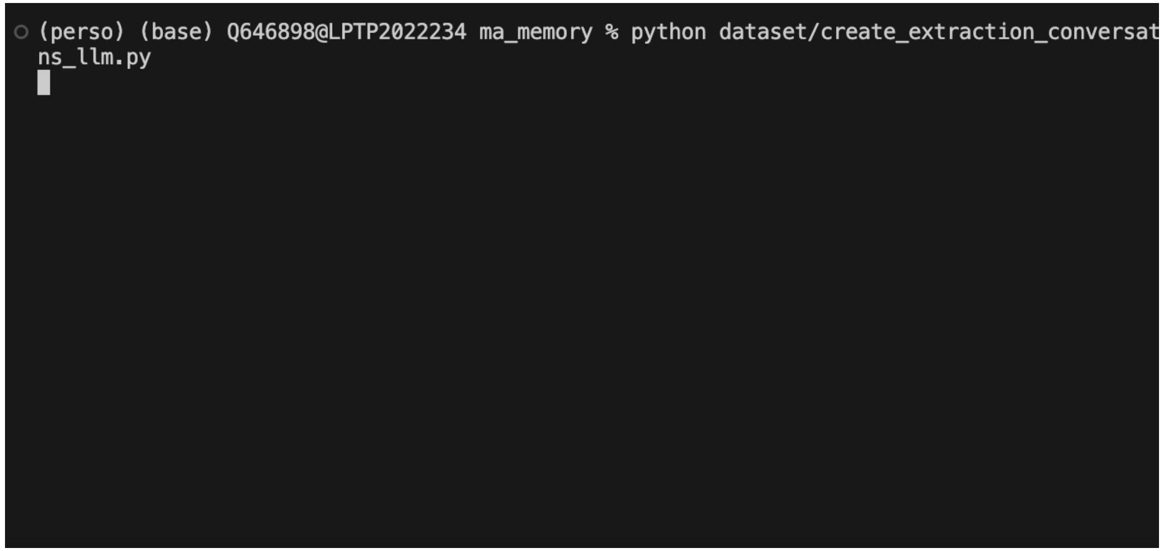
⋮

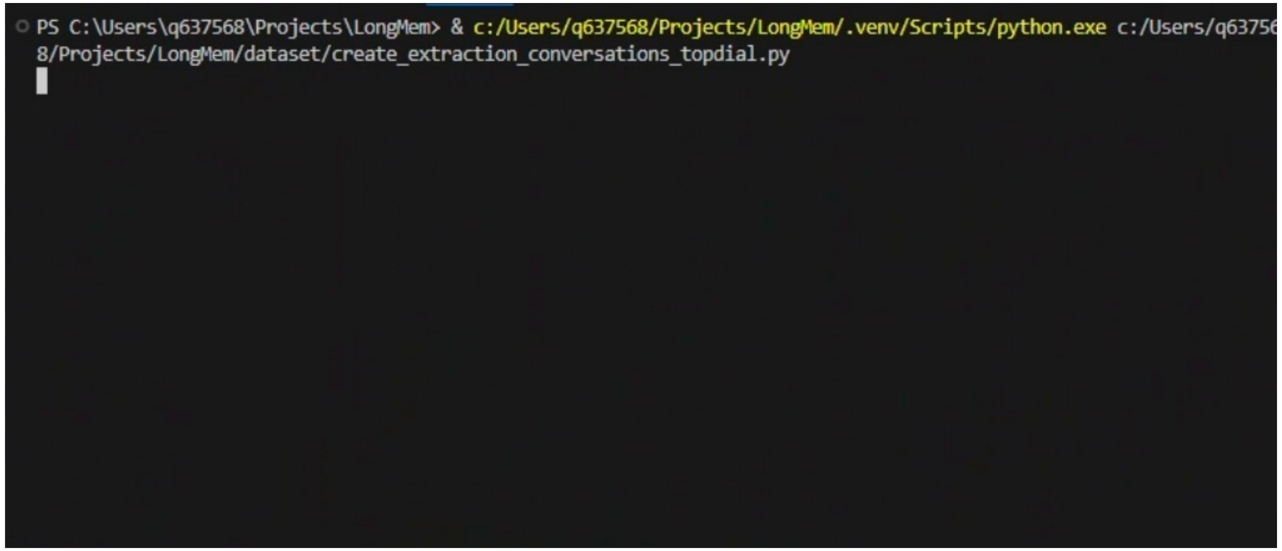
11. RQ1: Dataset Generation

Dataset – Generation Methods

LLM-based generation methods:

1. one LLM generates the whole conversation
2. two LLM agents (user/driver and voice assistant) simulate conversation





Valid/Invalid Extraction

In-Car Conversation:
USER: I've been craving Italian food, can ...
ASSISTANT: Sure, ...

Valid Extraction:

```
{
  "points_of_interest": {
    "restaurant": {
      "favourite_cuisine": [
        {
          "user_sentence_preference_revealed": "I've been craving italian food",
          "user_preference": "Italian"
        }]}}}
```

Invalid Extraction:

```
{
  "navigation_and_routing": {
    "restaurant": {
      "favourite_food":
        {
          "user_sentence_preference_revealed": "I've been craving italian food",
          "user_preference": "Italian"
        }}}}
```

Reason not valid: (1) `restaurant` not subcategory of `navigation_and_routing`, (2) no detail category/attribute `favourite_food`, (3) value of detail category not of type `list`.

Introduction of other category

```
class Restaurant(BaseModel):
    favourite_cuisine: ...
    ...
    other: Optional[List[OutputFormat]] = Field(default=[], description="Extract here if one is
          true: it is a suggestion of the assistant, it is a question rather than a preference,
         it is a temporary wish, no category match or different to examples.")
```

Query Classification

```
class PointsOfInterestClassification(BaseModel):
    restaurant: Optional[bool] = Field(default=False, description="'True' if the user's query
        is in the category 'Restaurant'. This includes query in the topics 'Favourite Cuisine',
         'Preferred Restaurant Type', 'Fast Food Preference', 'Desired Price Range', 'Dietary
        Preferences', 'Preferred Payment method'.")


class QueryClassificationFunctionOutput(BaseModel):
    points_of_interest: PointsOfInterestClassification = Field(default=None,
        description="If the user's query is in the category 'Points of Interest'. This includes
            query in the topics 'Restaurant', 'Gas Station', 'Charging Station', 'Grocery
           Shopping'.",)
```

```
expr = "||".join([f"user_name=='{user_uuid}' && main_category=='{main_category}' &&
    subcategory=='{subcategory}'" for subcategory in sub_classified])
```

Prompts

Prompt: In-Car Conversation (One-Pref)

Instructions:
You are an advanced dataset creation algorithm specialized in generating human-assistant dialogues.
Your current task is to craft a realistic role-playing conversations between an in-car voice ASSISTANT (AI) and a USER (Human) within the vehicle, focusing specifically on the topic of '{topic}'.
Throughout the conversation, the USER should reveal a particular preference '{attribute}' related to the topic. The preference must be the only intent meaning no other preferences.

Output Format: Craft the dialogue in a JSON format, as shown in the example below:
...

Criteria for the Conversation:
- Maintain a realistic in-car context simulating the criteria of USER (human) and ASSISTANT (AI).
- It should be clear that the revealed preference is a consistent user choice, rather than a temporary desire, the user preference should be '{preference_strength_modulation}'.
- The USER initiates the conversation, with subsequent turns alternating between USER and ASSISTANT.
- Important: The dialogue should consist of {conversation_length} turns in total, with the user preference disclosed at the {position_user_preference_in_conv}. turn.
- The sentences should be a realistic output from speech-to-text models, meaning they should exclude quotation marks and other non-spoken text elements.

USER Description:
- Topic: {topic} USER Preference:
ASSISTANT Description:
- ASSISTANT Characteristics: The ASSISTANT is Confident, Ingenious, Empowering, Trustworthy, Caring, Joyful, and Empathetic. Replies should be short, concise, and informative.
- ASSISTANT Capabilities: The ASSISTANT is aware of the car's location: '{car_location_city}', can perform searches for places, access navigation including traffic information, provide car-related information, and control various car functions (e.g., climate control, lighting, start radio/music/podcasts).
- The ASSISTANT answers directly in one turn meaning it cannot say 'please wait' or 'one moment please'.
- ASSISTANT Memory: The ASSISTANT does not have memory and cannot store user preferences.
- ASSISTANT Proactivity: The ASSISTANT's level of proactivity is '{level_of_proactivity_assistant}'. Example for 'high' proactivity (direct answer - no question from the ASSISTANT): [user: "find nearby restaurant", assistant: "I found the restaurants A,B,C"], Example for 'low' proactivity of the assistant (question from the ASSISTANT): [user: "find nearby restaurant", assistant: "What cuisine are you in the mood for?"].

Knowledge
The USER and ASSISTANT do not know the descriptions of each other. This includes that the ASSISTANT is unaware of the topic and is unaware of the user preference.

Examples
These are one-turn examples from real in-car conversations:
{few_shot_examples}
Only use them as inspiration of realistic dialogues.

The conversation will be evaluated as correct if
- it is realistic and natural,
- it contains no user preference

Remember: the inclusion of any other preference ('avoid toll roads', 'avoid heavy traffic', 'set temperature to ...') leads to the conversation being useless.

11. RQ1: Dataset Generation

Prompt: In-Car Conversation (No-Pref)

Instructions:
You are an advanced dataset creation algorithm specialized in generating human-assistant dialogues.
Your current task is to craft a realistic role-playing conversations between an in-car voice ASSISTANT (AI) and a USER (Human) within the vehicle, focusing specifically on the topic of '{topic}'.
It is important that the user does not reveal any preference throughout the conversation, it should be general about the topic.

Output Format:
Craft the dialogue in a JSON format, as shown in the example below:
...

Criteria for the Conversation:
- Maintain a realistic in-car context simulating the criteria of USER (human) and ASSISTANT (AI).
- The USER initiates the conversation, with subsequent turns alternating between USER and ASSISTANT.
- Important The dialogue should consist of {conversation_length} turns in total.
- The sentences should be a realistic output from speech-to-text models, meaning they should exclude quotation marks and other non-spoken text elements.

USER Description:
- USER Profile:: {user_profile}.
- USER Conversation Style:: {user_conversation_style}.
Ensure USER's dialogue aligns with the defined profile and conversation style.

ASSISTANT Description:
- ASSISTANT Characteristics: The ASSISTANT is Confident, Ingenious, Empowering, Trustworthy, Caring, Joyful, and Empathetic. Replies should be short, concise, and informative.
- ASSISTANT Capabilities: The ASSISTANT is aware of the car's location: '{car_location_city}', can perform searches for places, access navigation including traffic information, provide car-related information, and control various car functions (e.g., climate control, lighting, start radio/music/podcasts).
- The ASSISTANT answers directly in one turn meaning it cannot say 'please wait' or 'one moment please'.
- ASSISTANT Memory: The ASSISTANT does not have memory and cannot store user preferences.
- ASSISTANT Proactivity: The ASSISTANT's level of proactivity is '{level_of_proactivity_assistant}'. Example for 'high' proactivity (direct answer - no question from the ASSISTANT): [user: "find nearby restaurant", assistant: "I found the restaurants A,B,C"], Example for 'low' proactivity of the assistant (question from the ASSISTANT): [user: "find nearby restaurant", assistant: "What cuisine are you in the mood for?"].

Knowledge
The USER and ASSISTANT do not know the descriptions of each other. This includes that the ASSISTANT is unaware of the topic.

Examples
These are one-turn examples from real in-car conversations:
{few_shot_examples}
Only use them as inspiration of realistic dialogues.

The conversation will be evaluated as correct if
- it is realistic and natural,
- the user preference reveal is natural and not out-of-the-box.
- it contains only the provided preference

Remember: the inclusion of any other preference ('avoid toll roads', 'avoid heavy traffic', 'set temperature to ...') leads to the conversation being useless.

Pfeile inputs

Prompt: Dynamic Inputs

- **user_profile** = {
"Age": "10-90", "Gender": ["male", "female"], "Technological_Proficiency": ["low", "middle", "high"],
}

- **user_conversation_style** = [
"Keyword-only: direct, straight-to-the-point.",
"Commanding: straightforward, imperative sentences.",
"Questioning: seeking information, clarification.",
"Conversational: casual, human-like manner."]

- **car_location_city** = sample_random_world_city

- **level_of_assistant_proactivity** = ["medium", "high", "very high - no questions"]

- **preference_strength_modulation** = ["subtly hinted at", "clearly stated", "strongly emphasized"]

- **conversation_length** = 4-10 sentences

- **position_user_preference_in_conv** = 1- conversation_length

- **few_shot_example** = real conversation turn related to topic

Prompt: Preference Query

Following conversation happened in a car between the user and the in-car voice assistant:
Conversation:
{conversation}

Your task is to craft a next-conversation question of the USER (on another day) to test if the ASSISTANT has extracted and saved the user preference: {user_preference}.

Frame the question generally in the higher-level topic to avoid giving hints about the user preference.
Examples:
===
User Preference: Vehicle Settings and Comfort; Climate Control; Airflow Direction Preferences; Face
next_conversation_question_user: Please turn on the air conditioning.
===
User Preference: Navigation and Routing; Parking; Need for Handicapped Accessible Parking; Yes
next_conversation_question_user: Find a parking space near the city centre.
===

Use the conversation style '{user_conversation_style}'.

Output Format:
Valid json:
"'json
{{
"next_conversation_question_user": "..."

Prompt: Maintenance Query

```
Following preference is stored in a database: '{user_preference}'.

Your task is to craft 3 user queries for a conversation with an in-car voice assistant.
1. Equal: User query includes the same preference.
2. Negate: User query negates the exact preference '{attribute}' (permanent) without naming a different preference.
3. Different: User query includes different preference: '{different_preference}' (permanent).

Use the conversation style '{user_conversation_style}'. Do not reference the already stored preference. It must be clear that the attribute is meant for the detail category '{detail_category}'. Do not directly ask to update a preference.

Example:

Revealed Preference: I am vegetarian, please find a suitable restaurant.
1. Equal: Can you find a restaurant that serves vegetarian food.
2. Negate: Can you find a steak restaurant as I am not vegetarian.
3. Different: Can you find a restaurant that serves kosher food.

#### Output Format:
Valid json:
```json
{{
"question_equal_preference": "..."
"question_negate_preference": "..."
"question_different_preference": "..."
}}
```
```

Prompt: Real few shot examples per topic

- **Restaurant**
  1. **User:** Show me restaurants nearby; **Assistant:** I have found multiple destinations in the category Restaurants. Which one should I select?
  2. **User:** I'm really hungry right now; **Assistant:** I have found these destinations for the category restaurant. Which one should I select?
  3. **User:** Fast Food Breakfast; **Assistant:** I have only found the destination Razz's Breakfast Bar and Grill, 5307, Westfair Ave, Schofield, Wisconsin. Should I start the guidance?
- **Gas Station**
  1. **User:** Take me to the nearest gas station; **Assistant:** I have found these destinations in the category Gas station. Which one should I select?
  2. **User:** Find me a Shell gas station; **Assistant:** I have found multiple results for shell in the category gas station. Which one should I select?
  3. **User:** what is my fuel usage today; **Assistant:** The average consumption since the last time the trip data was reset is 6,9 l/100km. Are you interested in ways of reducing your consumption?
- **Charging Station(in public)**
  1. **User:** Take me to the nearest charging station; **Assistant:** I have found these destinations for the category Charging station. Which one should I select?
  2. **User:** Take me to HPC; **Assistant:** I have found multiple destinations in the category HPC. Which one should I select?
  3. **User:** Where can I charge my car; **Assistant:** I have found multiple destinations in the category charging station. Which one should I select?
- **Grocery Shopping**
  1. **User:** Navigate to grocery store; **Assistant:** I have found these destinations for the category Grocery store. Which one should I select?
  2. **User:** Any supermarkets in the area; **Assistant:** I have found multiple destinations in the category Supermarkets. Which one should I select?
  3. **User:** Navigate to the local market; **Assistant:** I have found multiple destinations in the category Market. Which one should I select?
- **Routing**
  1. **User:** navigate to work place; **Assistant:** No problem. Let's go to work.
  2. **User:** show me alternative routes; **Assistant:** You are on the fastest route. Here you can manually select an alternative route if you still want to change it.
  3. **User:** avoid highways; **Assistant:** I'm avoiding highways.

Prompt: Extraction

```
You are an expert extraction algorithm. Extract relevant user preferences of the user {user_name} in the following conversation.
Only extract if preference said or confirmed by the user user_name, never from the assistant.

Conversation:
{conversation}

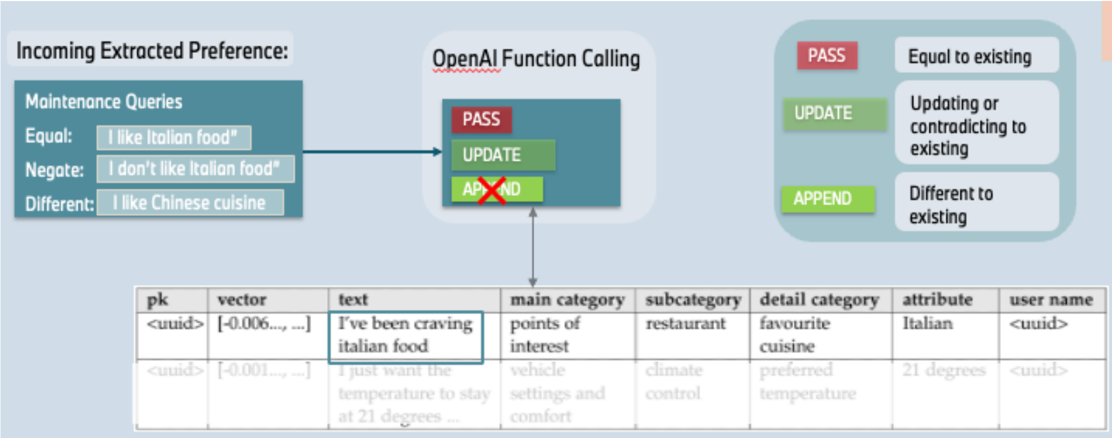
Only extract the preferences mentioned in the '{name_preference_function}' function, strictly follow the function parameters format.

If a preference is not present, do not include it in the output. If no preference, return null.

Custom Instructions: {custom_instructions}
```

13. RQ2: Preference Extraction + User Preference Storage

Prompt: Maintenance



- Maintenance Prompt Multiple Preferences Possible (MP):

("system", "You are a client to maintain a database storing user preferences. Your task is to perform a database function based on the incoming preference (focus on attribute) and existing preferences. You must call a tool. There are multiple preferences per category allowed. Examples: 1. (incoming: vegetarian, existing: vegetarian –> results in 'pass_preference'); 2. (incoming: vegetarian, existing: kosher –> results in 'append_preference'); 3. (incoming: vegetarian, existing: no vegetarian –> results in 'update_preference')."),
("human", "Existing Preferences: {existing_preferences} Incoming Preference: {incoming_preference}")

- Maintenance Prompt Multiple Preferences Not Possible (MNP)

("system", "...<same as mp prompt>... .There can always only be stored one preference."),
("human", "Existing Preferences: {existing_preferences} Incoming Preference: {incoming_preference}")

- **Pass**

```
class PassInput(BaseModel):
    incoming_preference: str = Field(description="the attribute of the incoming
        preference")
    pk_of_equal_existing_preference: str = Field(description="the primary key (pk) of
         the existing preference that is equal to the incoming preference attribute")

class Pass(BaseTool):
    name = "pass_preference"
    description = "call to pass incoming preference (not inserted in database) and
        keep existing preferences, perform if incoming preference attribute is equal
         or very similar to one existing preference attribute"
    args_schema: Type[BaseModel] = PassInput
```

- **Update:**

```
class UpdateInput(BaseModel):
    incoming_preference: str = Field(description="the attribute of the incoming
        preference")
    pk_of_to_delete_existing_preference: str = Field(description="the primary key (pk)
         of existing preference that should be deleted")

class Update(BaseTool):
    name = "update_preference"
    description = "call to delete one existing preference and insert incoming
        preference, perform if incoming preference attribute is updating or
        contradicting one existing preference attribute"
    args_schema: Type[BaseModel] = UpdateInput
```

- **Append:**

```
class AppendInput(BaseModel):
    incoming_preference: str = Field(description="the attribute of the incoming
        preference")

class Append(BaseTool):
    name = "append_preference"
    description = "call to append incoming preference to database and keep existing
        preferences, call if incoming preference attribute is different to existing
        preferences attributes, it can be of the same category"
    args_schema: Type[BaseModel] = AppendInput
```

15. RQ4: User Preference Storage Maintenance

Results

RQ1: Dataset

How can a suitable conversational dataset be created to develop and evaluate the personal preference memory system?

Dataset Statistics:

	In-Car Conversations	Preference Query	Maintenance Queries
LLM (Azure GPT-4 Turbo)			
Median tokens	976	353	357
Median prompt tokens	≈ 830	≈ 343	≈ 302
Median completion tokens	≈ 143	≈ 13	≈ 55
Cost per generation	≈ 0.01268$	≈ 0.00382$	≈ 0.00467$
Median Generation Time (P50)	16.26s	2.56s	5.29s
Output			
Mean conversation length	5.08		

1000 *One-Prefence* Conv.
1000 Preference Query
1000 Maintenance Query
100 *No-Preference* Conv.

~23$

Human Evaluation:

Criteria	Average Score [1,3]↑	Ratio 'Valid' [0,1]↑
In-Car Conversations		
Realism of User	2.73	
Realism of Assistant	2.93	
Organicness of User Preference	2.67	
Clarity of User Preference	2.47	
Environment Understanding	3.0	
Overall Subjective Quality	2.18	
Valid Conversation for Dataset		31/40 = 0.78
Preference Query		
Overall Subjective Quality	2.75	
Valid Question for Dataset		38/40 = 0.95
Maintenance Queries		
Overall Subjective Quality	2.71	
Valid Maintenance Questions for Dataset		40/40 = 1.0

Setting

3 Judges (Bmw Experts)

Subset of

- 40 data points (One-Pref)
- 5 data points (No-Pref)

6 x Preference not clear enough
3 x Multiple preferences revealed

Dataset Distribution

B.5. Dataset Distributions

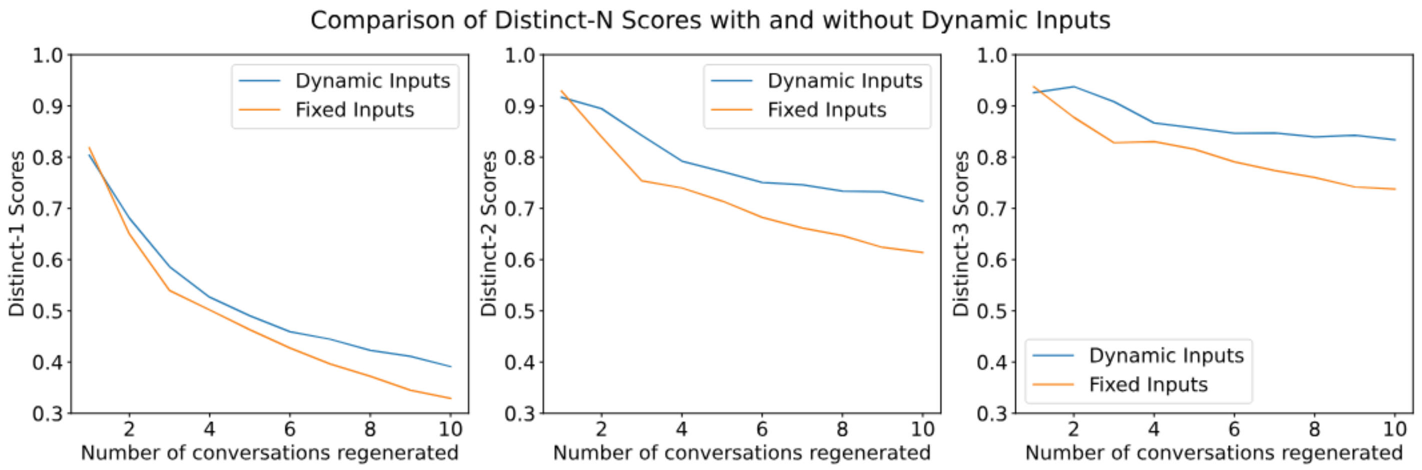
User Profiles:
Age: 30 (25), 80 (19), 60 (14), 50 (13), 40 (12), 20 (9), 70 (8).
Technological Proficiency: Middle (42), High (31), Low (27).
User Conversation Style: Keyword only (34), Conversational (26), Commanding (22), Questioning (18).

User Preferences:
Main Category: Points of Interest (384), Navigation and Routing (255), Entertainment and Media (214), Vehicle Settings and Comfort (147).
Subcategory: Parking (142), Restaurant (141), Radio and Podcasts (120), Charging Station (104), Music (94), Climate Control (81), Gas Station (80), Routing (75), Lighting and Ambience (66), Grocery Shopping (59), Traffic and Conditions (38).
Preference Type: MNP (579), MP (421).
Detail Category: Favorite Podcast Shows (37), Favorite Podcast Genres (35), Dietary Preferences (34), Charging Station Amenities (34), Preferred Supermarket Chains (33), Preferred Gas Station (31), General News Source (29), Favorite Songs (28), Tolerance for Traffic (28), Handicapped Parking (27), Preferred Charging Network (26), Covered Parking (26), Local Markets/Farms (26), Airflow Preferences (26), Green Fuel (26), Paid Parking Sensitivity (25), Favorite Cuisine (25), Favorite Genres (25), Shortest Time/Distance (24), Lighting Ambient (24), Fuel Price Sensitivity (23), Fast Food (23), Lighting Color (23), Distance to Walk (23), Avoidance of Road Types (23), Everyday Charging (22), Travel Charging (22), Favorite Artists (22), Restaurant Type (22), Parking with Security (21), Fan Speed (21), Parking Type (20), Longer Route to Avoid Traffic (19), Music Streaming Service (19), Temperature (19), Radio Station (19), Lighting Brightness (19), Traffic Info Source (19), Desired Price Range (19), Payment Method (18), Seat Heating (15).

Conversation Criteria

Preference Strength Modulation: Clearly Stated (342), Strongly Emphasized (334), Subtly Hinted (324).
Conversation Length: 8 (266), 6 (261), 2 (255), 4 (218).
Position of User Preference in Conversation: 1 (494), 3 (270), 5 (173), 7 (63).

Dataset Diversity & Statistics

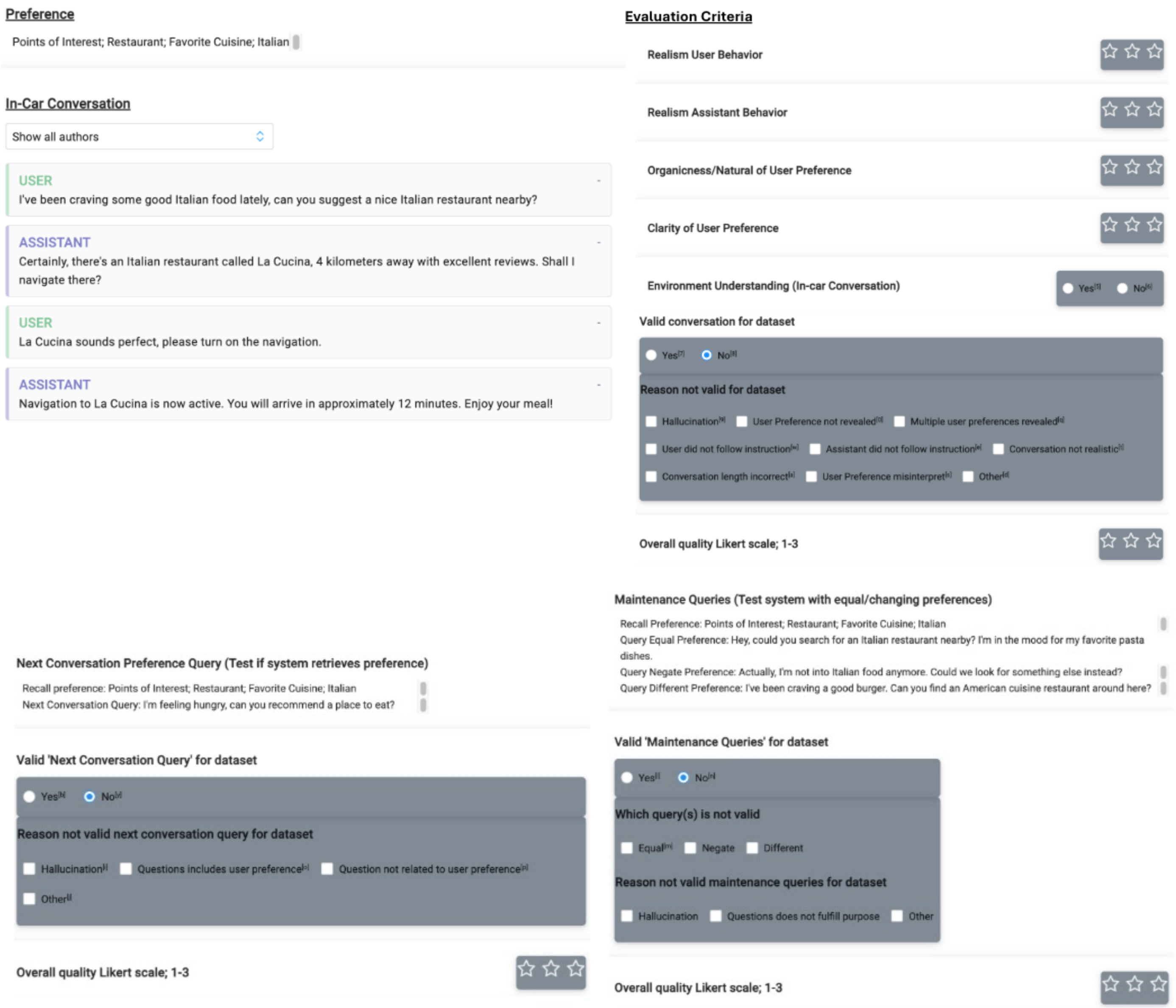


5 User
10 repeating conversations each

	In-Car Conversations	Preference Query	Maintenance Queries
LLM (Azure GPT-4 Turbo)			
Median tokens	976	353	357
Median prompt tokens	≈ 830	≈ 343	≈ 302
Median completion tokens	≈ 143	≈ 13	≈ 55
Cost per generation	≈ 0.01268$	≈ 0.00382$	≈ 0.00467$
Median Generation Time (P50)	16.26*s*	2.56*s*	5.29*s*
Output			
Mean conversation length	5.08		
Mean sentences per message	1.70		
Mean sentences per conversation	8.67		
Mean words per conversation	80.78		
Mean tokens per conversation	97.85		
Mean sentences per query		1.04	1.13
Mean words per query		8.34	12.06
Mean tokens per query		9.68	14.45

Table 5.1.: Dataset Statistics

Dataset Evaluation Interface & Criteria



1. **Realism of User Behavior**: Does the simulated user behave and speak in a manner that is consistent with how real users would act in a similar situation?
2. **Realism of Assistant Responses**: Are the assistant's responses appropriate, contextually relevant, and indicative of a sophisticated understanding of human speech patterns?
3. **Organicness of User Preference Revelation**: Is the user preference revealed in a way that feels organic to the conversation rather than forced or out of place?
4. **Clarity of User Preference**: How clearly is the user preference communicated within the conversation? Is it clear that it is a preference instead of a temporary wish?
5. **Environment Understanding**: Does the model demonstrate an understanding of the context in which the conversation is taking place (e.g., in-car environment)?

Valid 'Next Questions' for dataset:

- Answer **"No"** if:
  - Hallucination (LLM)
  - Questions includes user preference
  - Question not related to user preference
  - Other
- Else **„Yes"**

Valid 'Maintenance Questions' for dataset:

- Answer **"Yes"** if:
  - Question does what it should, i.e. be equal, negate, be different
- Else **„No"**

Dataset Evaluation Results No-Pref

Criteria	Average Score [1,3]↑	Ratio 'Valid' $[0, 1]$↑
In-Car Conversations		
Realism of User	3.0	
Realism of Assistant	3.0	
Environment Understanding	3.0	
Overall Subjective Quality	2.2	
Valid Conversation for Dataset		$5/5 = 1.0$

RQ2: Preference Extraction – Quantitative Evaluation

How could personal preferences be effectively extracted from conversations and stored?

Setting Evaluation on 400 One-Preference and 50 No-Preference *In-Car Conversations*

One-Preference Conv.

Category level	#categories	Accuracy		Precision		Recall		F1-Score		
		gpt-4-turbo	gpt-3.5-turbo	gpt-4-t.	gpt-3.5-t.	gpt-4-t.	gpt-3.5-t.	gpt-4-t.	gpt-3.5-t.	
Main (All)	4	0.92	0.76	0.94	0.94	0.98	0.79	0.96	0.86	
Sub (All)	11	0.86	0.71	0.89	0.88	0.97	0.76	0.93	0.82	**400**
Detail (All)	41	0.72	0.50	0.75	0.64	0.91	0.64	0.82	0.64	
Main (One-Pref.)	4	1.0	0.97	-	-	-	-	-	-	
Sub (One-Pref.)	11	0.99	0.94	-	-	-	-	-	-	**306**
Detail (One-Pref.)	41	0.94	0.77	-	-	-	-	-	-	

No-Preference Conv.

	Accuracy	#preferences extracted (orginal categories)	#preferences extracted (other category)
	gpt-4-turbo		
original categories	0.44	37	-
+ other category	0.68	19	14

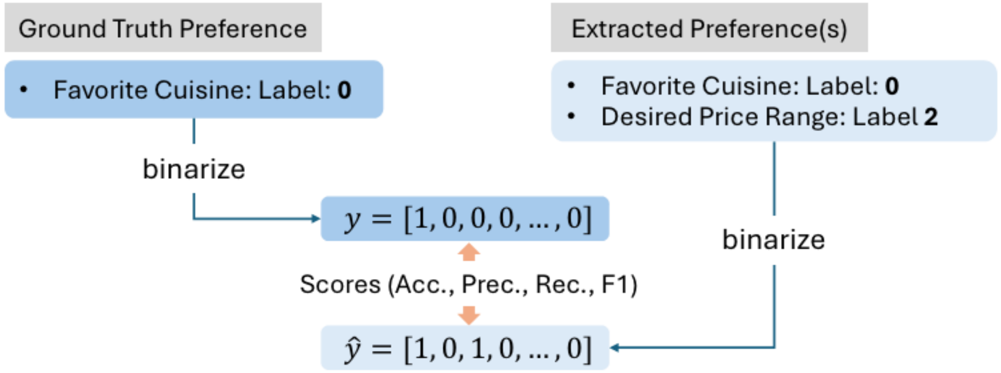
37. Introduction of other category

Findings

- Generated output adheres in 97% to the parameter schema
- Risk of non-extraction if a preference is present is low (2%)
- Risk of extraction if no preference is present is high (32%)
- Median tokens per extraction:
  - 8278 (7938 come from parameter schema)
  - Cost: 0.08$ per extraction

Result: Extraction

Vector Creation:



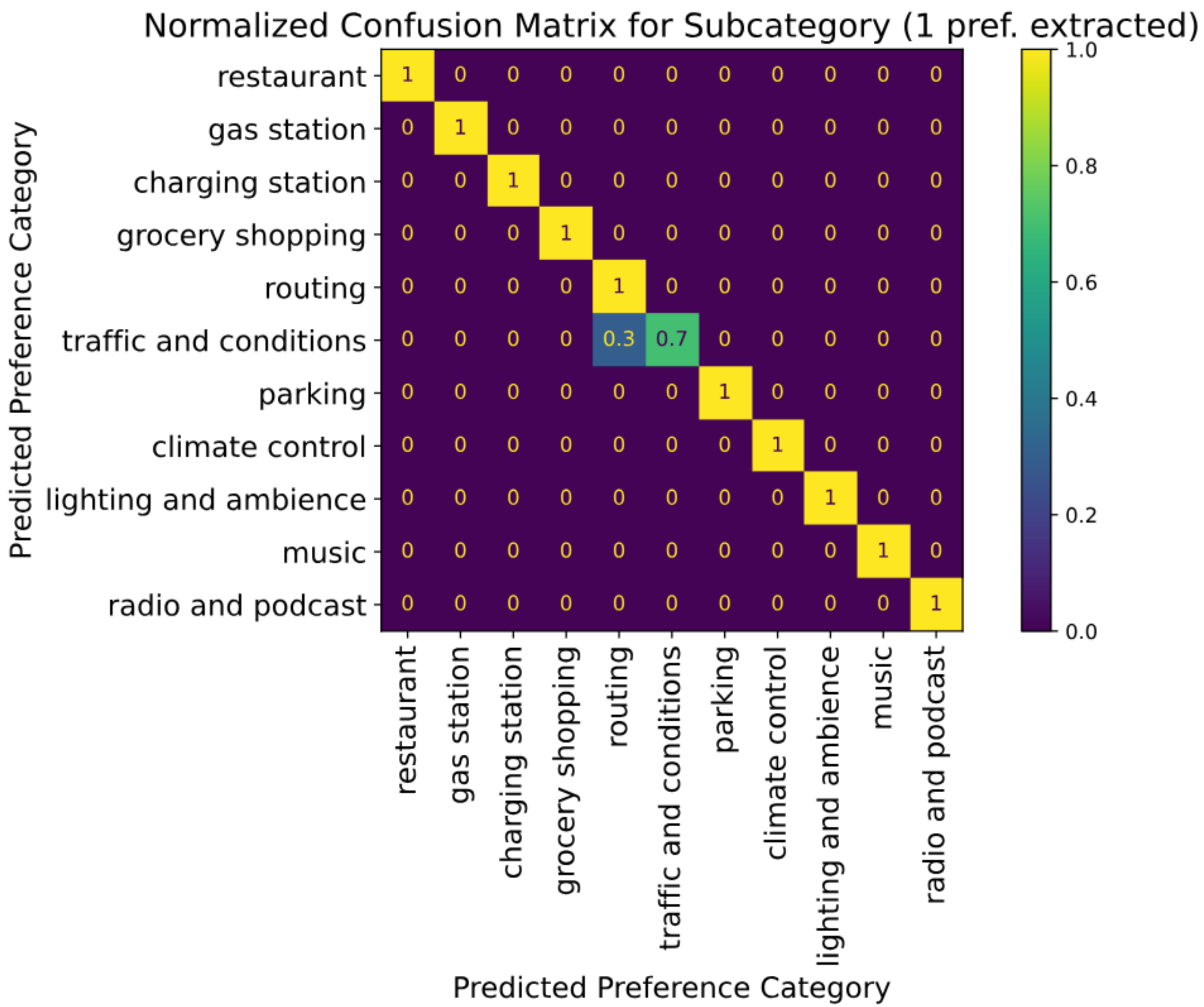
LLM		
Median tokens	8278	
of which prompt tokens	≈ 8178	
	of which ≈ 7938 function schema	
of which completion tokens	≈ 100	
	Azure GPT-4 Turbo	Azure GPT-3.5 Turbo
Cost per extraction	$\approx 0.0848\$$	$\approx 0.0042\$$
Median Latency (P50)	$7.85s$	$1.75s$

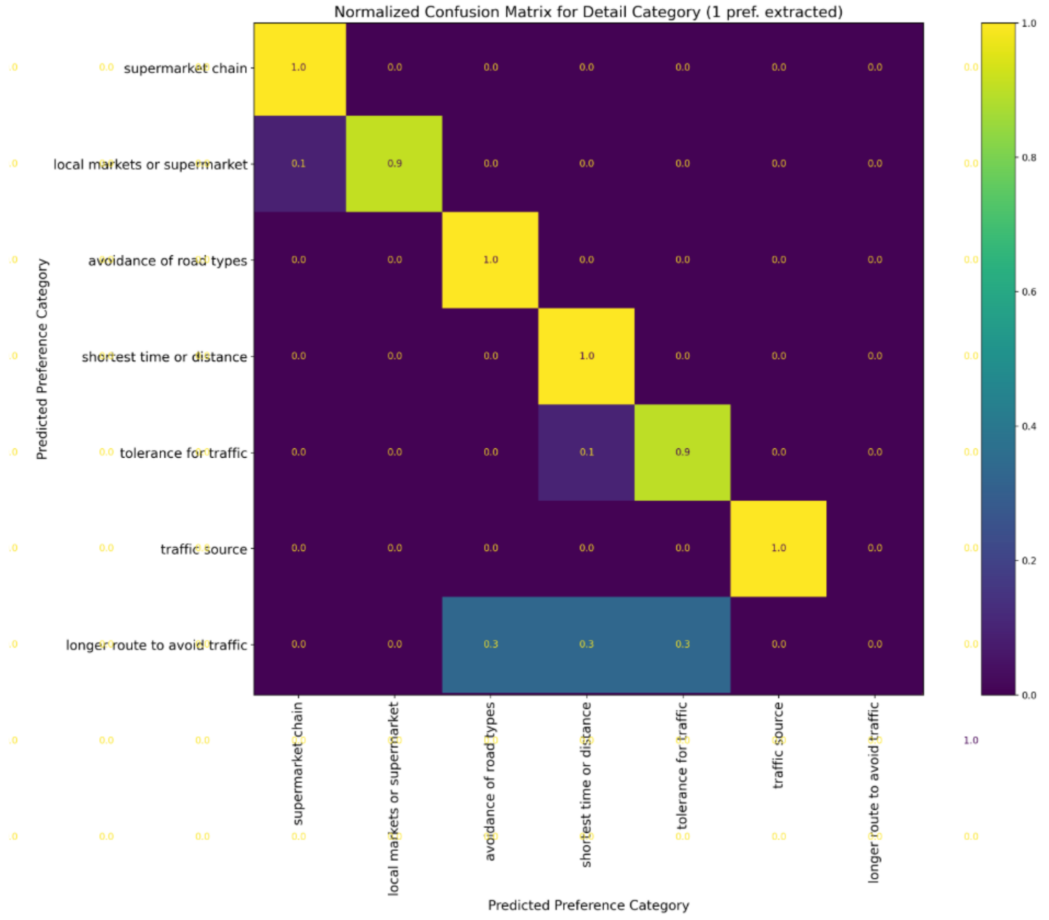
Table 5.4.: Preference Extraction Statistics

Of 400 preferences:	gpt-4-t.		gpt-3.5-t.	
	Count		Count	
no preference extracted	10	2.5%	80	20%
one preference extracted	306	76.5%	264	66%
multiple preferences extracted	84	21%	56	14%
valid (no validation error) at 1st try	387	96.8%	317	79.3%
valid (no validation error) at 2nd try	7	1.8%	6	1.5%
invalid extraction	6	1.5%	77	19.3%

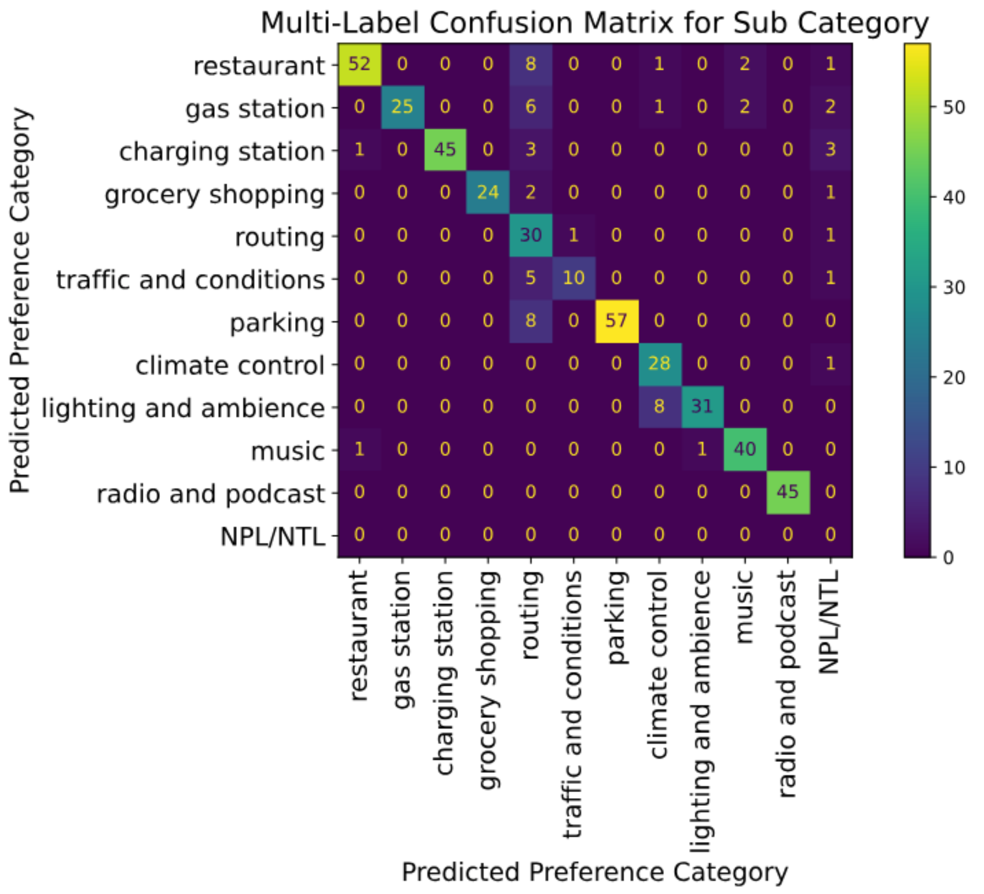
Table 5.5.: General performance of preference extraction based on 400 *In-Car-Conversations* of the *One-Preference* dataset.

Result: Extraction – Confusion Matrix





Extraction Multi-Label Confusion Matrix



e 5.8.: Multi-Label Confusion Matrix for the subcategories.

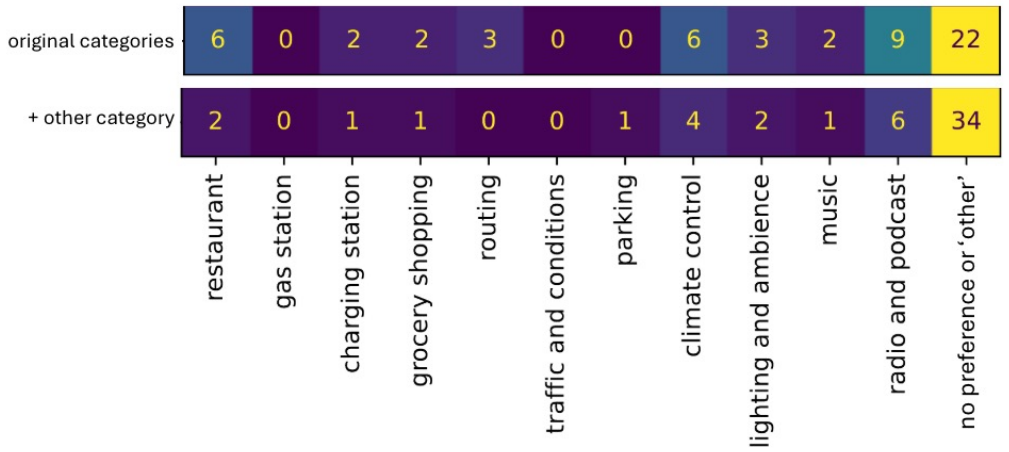


Figure 5.9.: Multi-Label Confusion Matrix for the subcategories.

RQ3: Preference Retrieval

Which methods enable a context-related retrieval of preferences?

Setting

- Evaluation on 289 data points (preferences extracted with perfect accuracy)
- *Preference Queries* target subcategory

Embedding-Based Retrieval

Embedding Approach	$\tilde{k} = n\_sub_{j,i}$		$\tilde{k} = n\_sub_{j,i} + 1$		$\tilde{k} = n\_sub_{j,i} + 2$	
	top-$\tilde{k}$ acc.	r.-prec.	top-$\tilde{k}$ acc.	r.-prec.	top-$\tilde{k}$ acc.	r.-prec.
EB-T	0.77	0.76	0.90	0.52	0.94	0.40
EB-DAT	**0.86**	**0.85**	**0.96**	**0.56**	**0.98**	**0.41**

$n\_sub_{j,i}$ ≙ number of preferences stored within subcategory i for user j

$$\text{r.prec.} = \frac{\#\,\text{preferences retrieved subcategory } i}{\#\,\text{total preferences subcategory } i}$$

Category-Based Retrieval

Category level	#categories	Accuracy		Precision		Recall		F1-Score		
		gpt-4-t.	gpt-3.5-t.	gpt-4-t.	gpt-3.5-t.	gpt-4-t.	gpt-3.5-t.	gpt-4-t.	gpt-3.5-t.	
Main (All)	4	0.90	0.80	0.92	0.94	0.98	0.85	0.95	0.88	**289**
Sub (All)	11	0.82	0.71	0.82	0.82	0.94	0.80	0.88	0.81	

→ relevance precision = 0.94

Query Classification

LLM		
Median tokens	1121	
of which prompt tokens	≈ 1110	
	of which ≈ 1020 function schema	
of which completion tokens	≈ 100	
	Azure GPT-4 Turbo	Azure GPT-3.5 Turbo
Cost per extraction	≈ 0.0142$	≈ 0.0007$
Median Latency (P50)	2.08s	0.48s

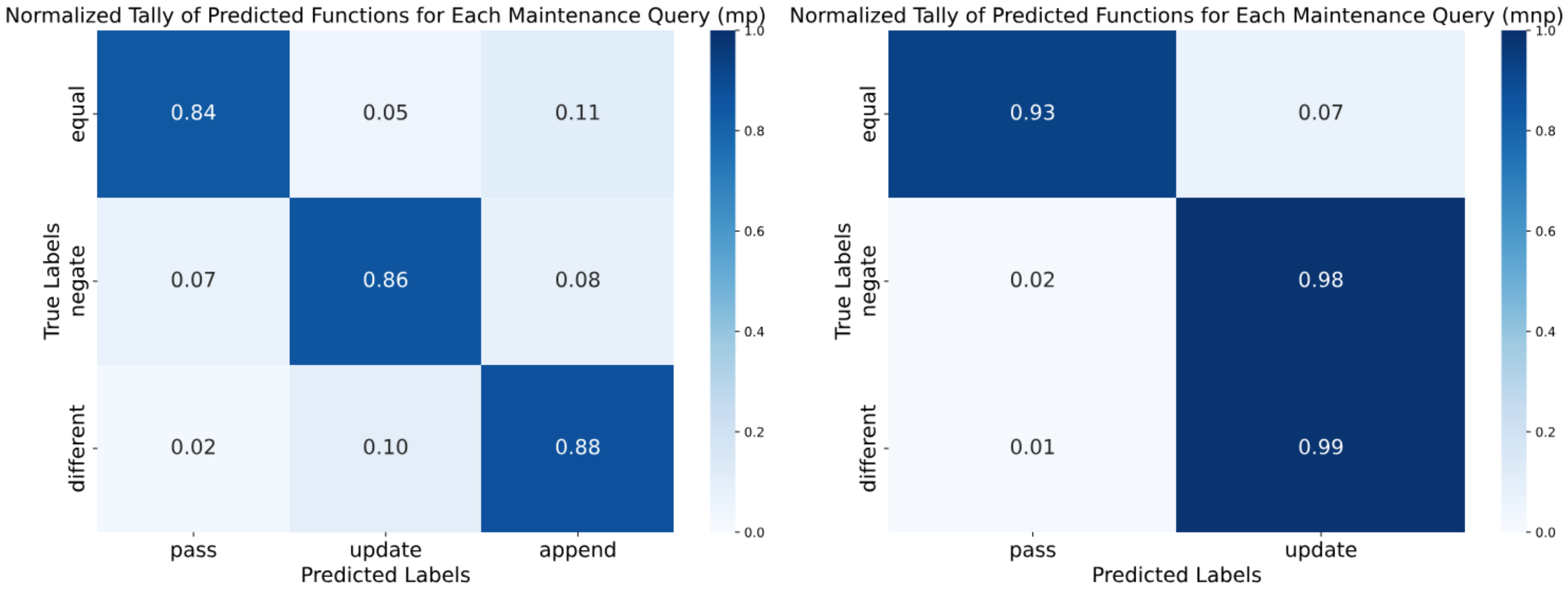
Table 5.9.: Query Classification Statistics

Of 289 preferences:	gpt-4-t.		gpt-3.5-t.	
	Count		Count	
no category classified	3	1.0%	38	13.1%
one category classified	251	86.8%	225	77.9%
multiple categories classified	35	12.1%	26	9.0%
valid (no validation error) at 1st try	288	99.7%	251	86.9%
valid (no validation error) at 2nd try	1	0.3%	1	0.3%
invalid extraction	0	0%	37	12.8%

RQ4: User Preference Storage Maintenance

Which method can be used to effectively maintain the preference storage when personal preferences change over time?

Setting	Evaluation on 829 *Maintenance Queries* • 289 equal preference • 251 negate preference • 289 different preference



(a) Multiple Preferences Possible (MP)

(b) Multiple Preferences Not Possible (MNP)

694 preferences less in storage than without maintenance

[1] Bae, S., Kwak, D., Kang, S., Lee, M. Y., Kim, S., Jeong, Y., Kim, H., Lee, S.-W., Park, W., & Sung, N. (2022). *Keep Me Updated! Memory Management in Long-term Conversations*. http://arxiv.org/abs/2210.08750

Maintenance Function Call

LLM		
Median tokens	415	
of which prompt tokens	≈ 370	
	of which ≈ 198 function schema	
of which completion tokens	≈ 45	
	Azure GPT-4 Turbo	Azure GPT-3.5 Turbo
Cost per maintenance	$\approx 0.0055\$$	$\approx 0.00028\$$
Median Latency (P50)	$2.82s$	$0.65s$

Table 5.13.: Maintenance Functions Call Statistics

Category Type	***Maintenance Query***		# data points	**Accuracy**	
				gpt-4-t.	gpt-3.5-t.
MP	equal	($\rightarrow$ *pass*)	128	0.84	0.67
	negate	($\rightarrow$ *update*)	118	0.86	0.72
	different	($\rightarrow$ *append*)	128	0.88	0.88
	average		374	0.86	0.76
MNP	equal	($\rightarrow$ *pass*)	161	0.93	0.75
	negate	($\rightarrow$ *update*)	133	0.98	0.98
	different	($\rightarrow$ *update*)	161	0.99	0.93
	average		455	0.97	0.89

Table 5.14.: Accuracy of maintenance function call.

Discussion

Dataset

Error Type	Definition
Off-Prompt	Generation is unrelated to or contradicts the prompt.
Instruction Error	Error arises due to incomplete, incorrect, or unclear instructions in the prompt, leading to unintended generations.
Interpretation Error	Discrepancy between the user's intended instruction or description and the LLM's interpretation of it.

Guidelines. Finally, following guidelines are identified:

- **Explain it to the LLM like it's a 5th year old**: Derived from the famous prompts for LLM's, it is also important to give the LLM detailed, but clear and concise instructions to achieve the intended output. This improves Off-Prompt Error, Interpretation Error, and Instruction Error.
- **Avoid negations**: As also researched by Truong et al. [60], we experienced that LLM's struggle with negations. Therefore we have rewritten many of our preference attributes, for instance: "Preference for Covered Parking: No" → "Preference for Covered Parking: Indifferent to Parking" which led to a significant decrease of Interpretation Error
- **Keep in mind:** LLM's are generative and hallucinate. Since there is no reliable method to detect hallucinations yet, a human in the loop would be needed for perfect quality.

Error Type	Example
Off-Prompt	**Preference not clear** (common - happened in 6/40 generated conversations): **Generated In-Car Conversation**: USER: Find VoltRise Charging station nearby. ASSISTANT: The nearest VoltRise Charging station is ... **Prompt:** ... It should be clear that the revealed preference is a consistent user choice, rather than a temporary desire, the user preference should be '{preference_strength_modulation}'. ... **Multiple preferences revealed** (common - happened in 3/40 generated conversations): **Generated In-Car Conversation**: USER: I only fill up at GasGlo stations. Can you find one nearby? ASSISTANT: I've located a GasGlo station 3.5 miles away. Would you like directions to it? USER: Yes, please start navigation, and no toll roads if that's an option. ASSISTANT: ... **Prompt:** ... Throughout the conversation, the USER should reveal a particular preference '{attribute}' related to the topic. The preference must be the only intent meaning no other preferences. ...
Instruction Error	**Attribute not descriptive enough standalone**: (rare) **Preference**: Detail Category: Price Sensitivity for Fuel, Attribute: Rather cheapest **Generated In-Car-Conversation**: USER: Find gas station, cheapest option. ASSISTANT: ... Not clear to the LLM that there is another preference within that detail category with attribute "cheapest". Will create issue for extraction evaluation.
Interpretation Error	**Attribute misinterpret:** (rare) **Preference**: Detail Category: Distance Willing to Walk from Parking to Destination, Attribute: not relevant (closest with low cost) **Generated In-Car-Conversation**: USER: Find parking, must be closest and cheapest. ASSISTANT: ... **Intended**: distance not relevant, find situational best option; **Misinterpret**: closest AND cheapest.

Extraction

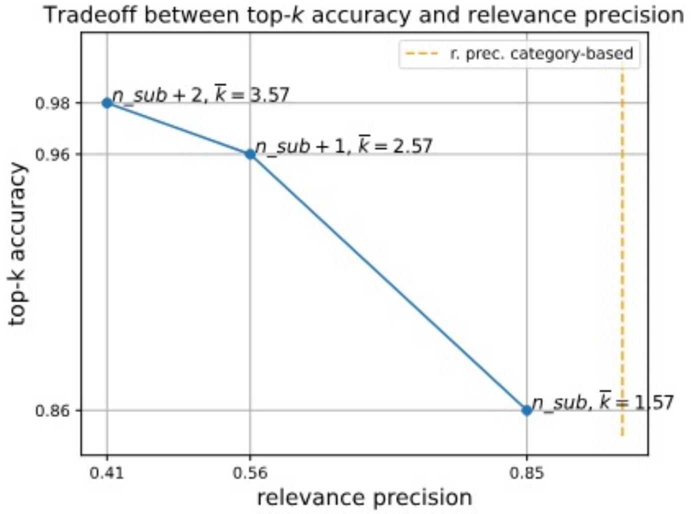
Error Type	Definition
Dataset-Induced Errors	Errors caused by the dataset, such as unclear or multiple preferences within a single conversation.
Off-Prompt Extraction	The LLM generates extraction output that is unrelated to or contradicts the given prompt.
Misinterpretation of Context	The LLM misinterprets the context or intent of the user's messages, leading to incorrect preference extraction. For instance when an user questions about the topic or a temporary wish get misinterpreted as a preference.
Invalid Output Structure	The LLM fails to adhere to the function schema and does not generate a valid output.
Hallucination of Preference	The LLM identifies preferences that were not present in the original conversation.

Table 6.3.: Overview of different error types based on our preference extraction with the function calling method.

Error Type	Example
Off-Prompt Extraction	**Assistant suggestion gets extracted as preference**: **In-Car Conversation**: USER: ... what kind of music stations are available around here? ASSISTANT: ... a variety of genres including classical, pop, and local music ... **Extracted Preference**: entertainment_and_media: radio_and_podcast: preferred_radio_station: local music station **Prompt**: ... Only extract if preference said or confirmed by the user {user_name}, never from the assistant. ...
Misinterpretation of Context	**User temporary wish identified as preference**: (rare) **In-Car-Conversation**: USER: Navigate to Bella's Italian Cafe ASSISTANT: Sure, setting navigation ... **Extracted Preference**: "points_of_interest": "restaurant", "preferred_restaurant_type", "Italian"
Invalid Extraction Output	**Output structure not conform with category schema:** (rare) **Extraction**: {"points_of_interest": {"points_of_interest": {"other": [{"user_sentence_preference_revealed": "Historic landmarks nearby", "user_preference": "historic landmarks"}]}}}
Hallucination of Preference	**In-Car Conversation**: USER: center Akhisar route ASSISTANT: Starting navigation to Akhisar city center. ... **Extracted Preference**: "navigation_and_routing": "routing": "avoidance_of_specific_road_types": "center"

Table 6.4.: Examples for the different error types based on our preference extraction with the function calling method.

Retrieval



- **Embedding-based** retrieval fails when
  - the *Preference Query* is semantically similar to additional preferences outside of the ground truth subcategory.
    - Example:
      - **Ground Truth Preference:**

```
{"main_category": "points_of_interest",
"subcategory": "restaurant"
"detail_category": "dietary_preference",
"attribute": "Nut Allergies",
"text": "I have a nut allergy."}
}
```

      - *Preference Query*: USER: Recommend a restaurant nearby.
      - **False retrieved preference:** {"main_category": "points_of_interest", "subcategory": "**grocery_shopping**", "detail_category": "preferred_supermarket_chain", "attribute": "FreshFare Hub", "text": "Always take me to FreshFare Hub for groceries"}
      - **Category-based** retrieval succeeded as query it fits neatly into the restaurant subcategory.

- **Category-based** retrieval fails
  - when the *Preference Query* does not fit neatly into a single subcategory.
    - Example:
      - **Ground Truth Preference:**

```
{"main_category": "entertainment_and_media",
"subcategory": "radio_and_podcast"
"detail_category": "favorite_podcast_genres",
"attribute": "Science",
"text": "I've always loved learning about new scientific discoveries."}
}
```

      - *Preference Query*: USER: What should we listen to today while driving?
      - **Misclassified Category:** {"entertainment_and_media": {"**music**": true}}
      - **Embedding-based** retrieval succeeded also because user had no preference within music.
  - because of general LLM-related issues (refer to error types for LLM-based extraction identified in subsection 6.2.1).

- **Both** methods fail when
  - the *Preference Query* is too ambiguous so that there exist more related categories and no specific semantic similar preference.
    - Example:
      - **Ground Truth Preference:**

```
{"main_category": "points_of_interest",
"subcategory": "charging_station"
"detail_category": "preferred_type_of_charging_while_traveling",
"attribute": "DC",
"text": "I always prefer the speed of DC charging when traveling."}
}
```

      - *Preference Query*: USER: Can you suggest a route for our trip this weekend?

Template

Stick to the good sebis traditions

- Provide action links at the bottom of the slide to guide the audience to our web pages or publications (see below). (Select the text, press CTRL-K)

- Use a file name according to our sebis conventions which helps us and our audience to find the file of your presentation on our web site with Google search:
  - YYMMDD Author Short Title
  - Include this string in the footer (Einfügen -> Kopf- und Fusszeile -> Fusszeile)
  - The unusual date format simplifies the search for the latest version of a slide in an alphabetical directory listing (Dropbox, Explorer, Tricia, Sky-Drive)

[Ha13g] Hauder, M., Roth, S., Matthes, F.: Current Tool support for Metrics in Enterprise Architecture Management

For more information visit BEAMS , EAM Pattern Catalog and EAM KPI Catalog (http://wwwmatthes.in.tum.de)

Use the sebis visual language (shapes, fonts, colors, sizes)