Beyond Single-User Dialogue: Assessing Multi-User Dialogue State Tracking Capabilities of Large Language Models

Sangmin Song[1*], **Juhwan Choi**[2*], **JungMin Yun**[3], and **YoungBin Kim**[3]
[1]GS Neotek [2]AITRICS [3]Chung-Ang University
[1]song0313@gsneotek.com [2]jhchoi@aitrics.com
[3]{cocoro357, ybkim85}@cau.ac.kr

Abstract

Large language models (LLMs) have demonstrated remarkable performance in zero-shot dialogue state tracking (DST), reducing the need for task-specific training. However, conventional DST benchmarks primarily focus on structured user-agent conversations, failing to capture the complexities of real-world multi-user interactions. In this study, we assess the robustness of LLMs in multi-user DST while minimizing dataset construction costs. Inspired by recent advances in LLM-based data annotation, we extend an existing DST dataset by generating utterances of a second user based on speech act theory. Our methodology systematically incorporates a second user's utterances into conversations, enabling a controlled evaluation of LLMs in multi-user settings. Experimental results reveal a significant performance drop compared to single-user DST, highlighting the limitations of current LLMs in extracting and tracking dialogue states amidst multiple speakers. Our findings emphasize the need for future research to enhance LLMs for multi-user DST scenarios, paving the way for more realistic and robust DST models.

1 Introduction

Large language models (LLMs) have achieved remarkable success across various natural language processing tasks, ranging from translation to mathematical problem-solving (Brown et al., 2020; Achiam et al., 2023; Dubey et al., 2024; Team et al., 2024; Hurst et al., 2024). Their effectiveness has also been demonstrated in dialogue state tracking (DST), a task that predicts a user's goal based on conversations between the user and an agent (Jacqmin et al., 2022). Traditional approaches to DST typically rely on training sequence-to-sequence models (Heck et al., 2020; Campagna et al., 2020; Li et al., 2021). However, recent studies suggest that LLMs can achieve competitive performance in DST through zero-shot inference, eliminating the need for task-specific training (Heck et al., 2023).

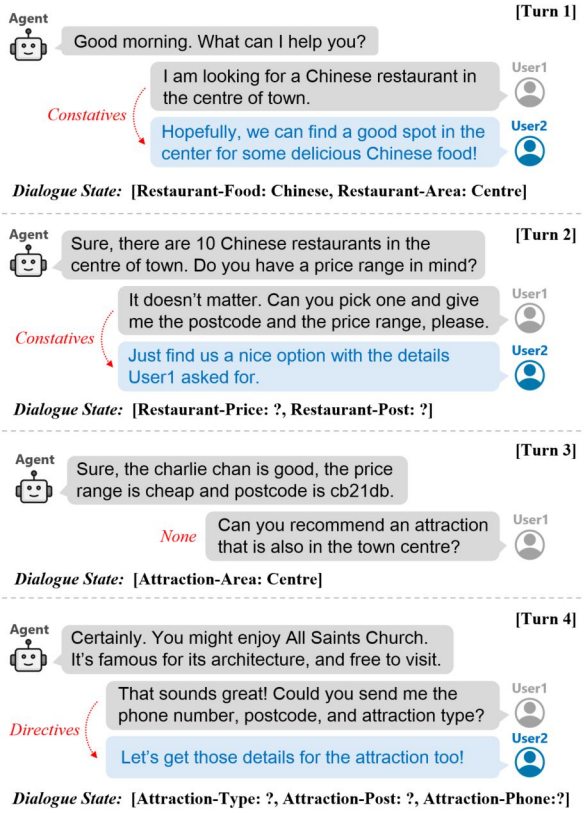

Figure 1: Example of a dialogue and slot values where the utterance of $user_2$ is introduced through our proposed method. We use this multi-user dialogue to evaluate LLM-based zero-shot DST approaches and compare their performance with existing single-user DST.

Despite these advances, conventional DST benchmarks primarily feature structured user-agent dialogues, limiting their applicability to more complex real-world scenarios. For instance, MultiWOZ (Budzianowski et al., 2018), one of the most widely used DST datasets, contains only single-user conversations. In contrast, real-world interactions often involve multiple users collaboratively engag-

*Equal contribution as co-first author.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 20018–20029

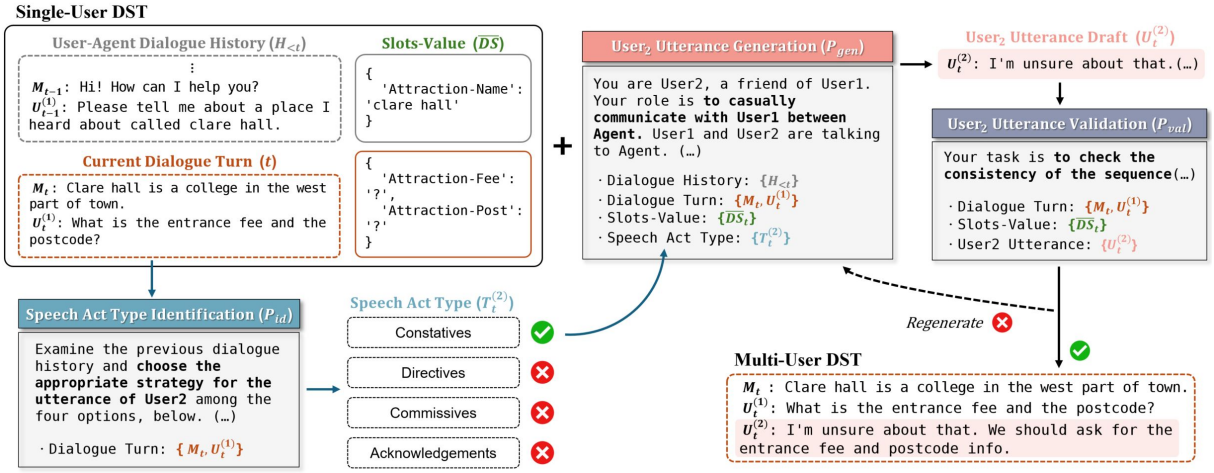

Figure 2: The illustration of our method stated in Section 2.2.

ing in decision-making before interacting with an agent. For example, a group of friends discussing restaurant options might first debate their cuisine preferences and locations before making a request to a booking system (e.g., "I prefer Korean food." → "Is there a good Korean restaurant in Suzhou?"). While previous studies introduced the concept of multi-user DST, they approached it by aggregating users' utterances and performing DST on the aggregated input as if it were a single-user setup (Jo et al., 2023). This highlights the need to assess DST models in genuinely multi-user scenarios to better evaluate their robustness and generalizability.

In this work, we conduct a case study to assess the performance of LLMs in multi-user DST, with minimal dataset construction costs. Inspired by recent efforts to employ LLMs for data annotation, we extend existing DST datasets using state-of-the-art LLMs (Tan et al., 2024; Choi et al., 2024). Specifically, we generate additional user utterances that interleave between the original user-agent exchanges. Our approach comprises two main steps: (1) identifying the appropriate type of interjections based on speech act theory (Austin, 1975; Bach and Harnish, 1979), and (2) prompting an LLM to generate and validate the second user's utterances accordingly. Figure 1 presents an example of a dialogue with utterances of the second user, based on our suggested framework. We compare the performance of LLMs in cases where the second user utterance is present or absent throughout the experiment.

Experimental results on our extended dataset show a substantial performance drop compared to single-user DST. This indicates that current LLMs struggle to correctly extract and track dialogue states in conversations with multiple users, underscoring limitations in their generalizability. Our findings highlight the need for future research to develop DST methods and prompt-engineering strategies optimized for multi-user settings.

Our primary contributions are as follows:

- We propose the first systematic evaluation of LLMs in multi-user DST, highlighting the unique challenges posed by multi-user interactions.
- We introduce a cost-efficient method to extend existing DST datasets by generating additional user utterances based on speech acts using state-of-the-art LLMs.
- We conduct comprehensive experiments demonstrating a notable performance drop in multi-user DST, underscoring the limitations of current LLMs when handling complex dialogue dynamics.

2 Methodology

2.1 Preliminaries

In this subsection, we briefly explain preliminaries regarding DST, especially focusing on zero-shot DST using LLMs. We follow the definitions and notations of a previous study (Heck et al., 2023).

DST is a core component of task-oriented dialogue systems, responsible for tracking the user's evolving intent throughout a conversation. At each dialogue turn t, given a user utterance U_t and a preceding agent utterance M_t, DST predicts a structured set of slot-value pairs representing the

user's goal. We call this structured set of slot-value pairs the dialogue state DS_t, which encapsulates the user's intent and relevant context at turn t. DS_t is updated as follows:

$$DS_t = DS_{t-1} \cup \widehat{DS}_t$$

where $\widehat{DS}_t$ is the newly inferred slot-value set at turn t.

Traditional DST models require extensive supervised training on labeled datasets (Heck et al., 2020; Campagna et al., 2020; Li et al., 2021). However, recent advances in LLMs have enabled zero-shot DST, where a general-purpose model extracts slot values without explicit fine-tuning (Feng et al., 2023; Heck et al., 2023). In zero-shot DST, the inference-time slot set $S = \{(s_1, v_1), (s_2, v_2), ..., (s_n, v_n)\}$, where s_i denotes a slot and v_i its corresponding value, is disjoint from the training-time slot set S', meaning $S \cap S' = \emptyset$. This requires the model to generalize without prior domain-specific training.

Zero-shot DST using LLMs typically follows a prompting-based approach. P, an instruction including a task description, is provided in natural language, guiding the model to extract relevant slot-value pairs from a conversation. At each turn, the prompt A_t includes both agent and user utterances:

$$A_1 = P \oplus \text{"agent": } M_1 \oplus \text{"user1": } U_1^{(1)},$$
$$A_t = \text{"agent": } M_t \oplus \text{"user1": } U_t^{(1)}, \quad \forall t \in [2, T]$$

Unlike traditional approaches, LLMs perform DST by leveraging their conversational alignment and pre-trained world knowledge without modifying their parameters. In this study, we aim to validate the capabilities of LLMs in handling a multi-user DST scenario.

2.2 Construction of Multi-user DST Dataset

This subsection outlines our method to extend the existing dataset to incorporate a second user, thereby establishing multi-user DST. To accomplish this, we follow the concept of speech act theory (Austin, 1975; Bach and Harnish, 1979), which defines the types of human utterances.

We adopt four speech act types based on an established study (Bach and Harnish, 1979):

- **Constatives**: Statements that express factual information or claims, including answering, confirming, denying, disagreeing, or stating.
- **Directives**: Attempts to influence the actions of the addressee, such as advising, asking, requesting, forbidding, inviting, or ordering.
- **Commissives**: Utterances that commit the speaker to a future action, including promising, planning, vowing, betting, or opposing.
- **Acknowledgments**: Expressions of the speaker's attitude toward the hearer's social actions, such as apologizing, greeting, thanking, or accepting acknowledgments.

Based on these speech act types, we introduce user$_2$ utterances ($U_t^{(2)}$) into existing user-agent dialogues. Specifically, we insert $U_t^{(2)}$ at each dialogue turn t, within the original user1 utterance ($U_t^{(1)}$) and the agent response (M_t). To achieve this, we first use an LLM to determine the appropriate speech act type for $U_t^{(2)}$. This classification is represented as $T_t^{(2)} = \mathcal{L}(\mathcal{P}_{id}, U_t^{(1)}, M_t)$, where $\mathcal{L}$ denotes the LLM and $\mathcal{P}_{id}$ represents the corresponding prompt.

Once the speech act type is identified, we generate $U_t^{(2)}$ by prompting the LLM with relevant context, including $T_t^{(2)}$, $U_t^{(1)}$, M_t, dialogue history ($H_{<t}$), and the gold slot value ($\overline{DS}_t$). This process is formulated as $U_t^{(2)} = \mathcal{L}(\mathcal{P}_{gen}(T_t^{(2)}), H_{<t}, U_t^{(1)}, M_t, \overline{DS}_t)$.

To ensure the generated $U_t^{(2)}$ maintains conversation coherence and does not introduce hallucinated slot values, we conduct a validation step. If $V_t = \mathcal{L}(\mathcal{P}_{val}, U_t^{(1)}, U_t^{(2)}, M_t, \overline{DS}_t)$ returns True, we accept $U_t^{(2)}$ and insert it into the dialogue. Otherwise, we regenerate and revalidate $U_t^{(2)}$ up to three times. If validation fails in all attempts, we discard $U_t^{(2)}$ by setting it to $\emptyset$, ensuring that its inclusion does not compromise dialogue integrity.

This procedure is applied iteratively across all dialogue turns, extending the dataset to a multi-user DST setting. When performing DST on the extended dataset, we append user$_2$'s utterance to the prompt A_t as follows:

$$A'_t = A_t \oplus \text{"user2": } U_t^{(2)}$$
$$\text{if } U_t^{(2)} \neq \emptyset, \quad \forall t \in [1, T]$$

3 Experiment

3.1 Experimental Design

We introduce our experimental design to assess the performance of LLMs on multi-user DST using the extended dataset.

Evaluation Models. We adopt six different LLMs for experiments: GPT-4o (Hurst et al., 2024), o4-mini (OpenAI, 2025), Claude 3.5 Sonnet (Anthropic, 2024), Gemini-2.0-Flash (Pichai et al., 2024), LLaMA-3.1-8B (Dubey et al., 2024), and Gemma-2-9B (Team et al., 2024). These models are selected to evaluate performance across different pretraining paradigms and parameter sizes. Further details can be found in Appendix B.

Zero-Shot DST Setup. We use a prompting-based approach following (Heck et al., 2023). The prompt provides a task description, slot definitions, and the current dialogue history. At each turn, models extract slot-value pairs based on their internal knowledge. We keep inference parameters (temperature, top-p) at default values to ensure consistency across all models. A detailed breakdown of the prompt format is provided in Appendix G.4.

Evaluation Metric. To assess the performance of each model, we use Joint Goal Accuracy (JGA) (Henderson et al., 2014), which measures whether all slot values at each turn match the ground truth:

$$JGA = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(\widehat{DS}_t = \overline{DS}_t) \quad (1)$$

where N is the total number of turns, $\widehat{DS}_t$ is the predicted dialogue state, and $\overline{DS}_t$ is the ground truth. We measure JGA of each model on the original MultiWOZ 2.1 (Eric et al., 2020) and its extended version based on our methodology as stated in Section 2.2, thereby comparing the performance of the LLM on single-user and multi-user scenarios. Please refer to Appendix A for the statistics of the extended dataset. By analyzing the performance gap between these settings, we quantify the impact of the existence of $U_t^{(2)}$ on DST performance. Specifically, we hypothesize that LLMs will show a performance drop in a multi-user DST, even if the topic, context, and slot value of the dialogue remain identical.

3.2 Result

Table 1 presents a detailed performance comparison of various LLMs under both single-user and extended multi-user DST settings. Across all evaluated models, we observe a consistent drop in JGA when additional user_2 utterances are present, thus confirming our hypothesis that the presence of multiple speakers introduces added complexity. In particular, models such as GPT-4o and Claude 3.5 Sonnet experience more severe degradations (e.g., up to -7.1% in the train domain), indicating that even state-of-the-art LLMs struggle to disentangle the intertwined dialogue cues from multiple users. Notably, a similar drop in performance is observed in o4-mini, a reasoning LLM, implying that simply adopting reasoning LLMs may not provide a straightforward solution for multi-user DST, especially when accounting for their higher cost from increased token usage.

Models	Attr.	Hotel	Rest.	Taxi	Train	Avg.
GPT-4o	56.8	46.0	55.1	69.3	61.9	57.82
w/ user_2 utterances	-4.6	-2.1	-2.2	-1.7	-7.1	-3.54
o4-mini	56.3	44.9	52.6	66.8	63.5	56.82
w/ user_2 utterances	-3.6	-3.9	-3.5	-1.8	-1.1	-2.61
Claude 3.5 Sonnet	64.9	47.5	55.8	68.2	74.0	62.08
w/ user_2 utterances	-3.8	-1.9	-2.6	-1.3	-6.7	-3.26
Gemini-2.0-Flash	36.7	25.3	24.4	61.9	28.0	35.26
w/ user_2 utterances	-2.2	-2.5	-3.1	-1.2	-2.6	-2.32
LLaMA-3.1-8B	25.4	23.2	18.8	46.9	25.5	27.96
w/ user_2 utterances	-1.6	-0.3	-1.2	-0.4	-1.8	-1.06
Gemma-2-9B	40.0	36.3	44.4	61.5	35.4	43.52
w/ user_2 utterances	-1.1	-0.2	-2.0	-1.1	-2.4	-1.36

Table 1: Performance comparison between LLM models for zero-shot DST in per-domain JGA. The first row of each model represents the performance of the model on the original dataset (i.e., no user_2 utterances), and the second row denotes the performance degradation of the model on the extended dataset with $U_t^{(2)}$ compared to the original dataset. Note that Attr. and Rest. denote attraction and restaurant, respectively. Please refer to Appendix D for slot-level evaluation.

Our qualitative analysis in Appendix F attributes these drops to two main error types: (1) misextraction of slot values when user_2 utterances distract the model and (2) incorrect inference of slot types due to unintended slot predictions. These findings emphasize the need for more robust, speaker-aware DST strategies. Future work should explore sophisticated prompt engineering or model architectures to distinguish multiple speakers.

4 Conclusion

We presented a systematic investigation of multi-user DST, a task that has been largely overlooked in traditional DST research. By extending an ex-

isting dataset through LLM-based generation of additional $user_2$ utterances, we revealed that contemporary LLMs face notable performance drops in multi-user DST scenarios, even when the underlying user request remains unchanged. Our findings underscore the importance of focusing on multi-user dialogue and highlight the challenges in maintaining accurate slot tracking when multiple speakers contribute to the conversation.

Future work could explore more advanced prompt-engineering strategies or model designs that disentangle each speaker's goal. In real-world applications, multiple people often share or refine a collective plan, such as making group travel arrangements or ordering meals. Improving LLM-based DST for these more realistic, collaborative settings will be crucial for developing robust, universally applicable dialogue systems.

Limitations

While our methodology offers a straightforward means to extend a single-user dataset to a multi-user dialogue scenario, it does not fully capture the dynamic and often collaborative nature of real-world multi-user interactions. Below, we elaborate on the key limitations:

Limited Conversation Realistic and Dynamics. Our approach places additional speaker utterances into existing user-agent conversations without substantially altering the dialogue flow or the user's underlying intent, to accurately evaluate the effect of the additional utterance. In realistic multi-user settings, however, users often negotiate, question, and refine each other's ideas in ways that can change the system's ultimate goals. For example, two users might disagree on a restaurant's location or cuisine. Our current method does not model these forms of dynamic interaction, which may be less realistic and underrepresent more complex use cases. Nonetheless, it is important to note that current relatively minor additions of multi-user interactions bring significant performance degradation, which suggests that more realistic interactions would likely pose even greater challenges, further highlighting the need for research in this field.

Quality of Generated Utterances. We rely on LLMs to generate and validate the additional users' utterances. Although we employ validation steps to filter out incoherent or hallucinatory utterances, the generated data may still not fully reflect natural human conversational patterns. Furthermore, because these inserted utterances are not human-annotated, there is a risk that subtle contextual nuances are lost, and some inadvertently introduced biases or errors could remain unchecked. To precisely examine the quality of the generated utterances, we performed a manual inspection across three dimensions, which is presented in Appendix E.

Focus on Zero-shot DST. We specifically target zero-shot DST based on LLMs to highlight the inherent challenge posed by multi-user DST. While this isolates the effect of additional user utterances, it also may not reflect performance under scenarios where LLMs are not used. However, from our initial attempt to conduct experiments using existing DST methods that do not use LLMs, we found that they were not capable of handling this new multi-user interaction structure effectively. Accordingly, future research could investigate whether non-LLM or fully fine-tuned approaches better mitigate these multi-user complexities.

Overall, while our study demonstrates the feasibility of extending single-user dialogue datasets to multi-user scenarios at low cost, it serves primarily as a proof of concept to validate the limitations of current LLMs for multi-user DST, which is the key objective of this paper. Further methodological refinements in future studies, including human validation, richer conversation modeling, and domain diversification, will be essential to capture the true depth and breadth of multi-user task-oriented dialogues.

Acknowledgements

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [RS-2021-II211341, Artificial Intelligence Graduate School Program (Chung-Ang University)] and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2025-00556246).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Anthropic. 2024. Introducing claude 3.5 sonnet. Accessed: 2025-02-08.

John Langshaw Austin. 1975. *How to do things with words*. Harvard University Press.

Kent Bach and Robert M. Harnish. 1979. *Linguistic Communication and Speech Acts*. MIT Press.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Proceedings of NeurIPS*, pages 1877–1901.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of EMNLP*, pages 5016–5026.

Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica Lam. 2020. Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking. In *Proceedings of ACL*, pages 122–132.

Juhwan Choi, Jungmin Yun, Kyohoon Jin, and Young-Bin Kim. 2024. Multi-news+: Cost-efficient dataset cleansing via llm-based data annotation. In *Proceedings of EMNLP*, pages 15–29.

Suvodip Dey, Ramamohan Kummara, and Maunendra Desarkar. 2022. Towards fair evaluation of dialogue state tracking by flexible incorporation of turn-level performances. In *Proceedings of ACL*, pages 318–324.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceedings of LREC*, pages 422–428.

Yujie Feng, Zexin Lu, Bo Liu, Liming Zhan, and Xiao-Ming Wu. 2023. Towards llm-driven dialogue state tracking. In *Proceedings of EMNLP*, pages 739–755.

Michael Heck, Nurul Lubis, Benjamin Ruppik, Renato Vukovic, Shutong Feng, Christian Geishauser, Hsien-Chin Lin, Carel van Niekerk, and Milica Gasic. 2023. Chatgpt for zero-shot dialogue state tracking: A solution or an opportunity? In *Proceedings of ACL*, pages 936–950.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. Trippy: A triple copy strategy for value independent neural dialog state tracking. In *Proceedings of SIGDIAL*, pages 35–44.

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014. The second dialog state tracking challenge. In *Proceedings of SIGDIAL*, pages 263–272.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Léo Jacqmin, Lina M Rojas Barahona, and Benoit Favre. 2022. "do you follow me?": A survey of recent approaches in dialogue state tracking. In *Proceedings of SIGDIAL*, pages 336–350.

Yohan Jo, Xinyan Zhao, Arijit Biswas, Nikoletta Basiou, Vincent Auvray, Nikolaos Malandrakis, Angeliki Metallinou, and Alexandros Potamianos. 2023. Multi-user multiwoz: Task-oriented dialogues among multiple users. In *Findings of EMNLP*, pages 3237–3269.

Shuyang Li, Jin Cao, Mukund Sridhar, Henghui Zhu, Shang-Wen Li, Wael Hamza, and Julian McAuley. 2021. Zero-shot generalization in dialog state tracking through generative question answering. In *Proceedings of EACL*, pages 1063–1074.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of ACL*, pages 8086–8098.

Xiang Luo, Zhiwen Tang, Jin Wang, and Xuejie Zhang. 2024. Zero-shot cross-domain dialogue state tracking via dual low-rank adaptation. In *Proceedings of ACL*, pages 5746–5765.

OpenAI. 2025. Openai o3 and o4-mini system card. *OpenAI Blog*.

Sundar Pichai, Demis Hassabis, and Koray Kavukcuoglu. 2024. Introducing gemini 2.0: our new ai model for the agentic era. Accessed: 2025-02-08.

Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. Large language models for data annotation and synthesis: A survey. In *Proceedings of EMNLP*, pages 930–957.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Haoming Wang and Wang Xin. 2022. How to stop an avalanche? jodem: Joint decision making through compare and contrast for dialog state tracking. In *Findings of EMNLP*, pages 7030–7041.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of ACL*, pages 808–819.

Longfei Yang, Jiyi Li, Sheng Li, and Takahiro Shinozaki. 2023. Multi-domain dialogue state tracking with disentangled domain-slot attention. In *Findings of ACL*, pages 4928–4938.

A Dataset Statistics

In this section, we provide a detailed analysis of the dataset constructed for evaluating multi-user DST based on the aforementioned methodology. Our dataset is derived from the test set of MultiWOZ 2.1 (Eric et al., 2020), an existing DST benchmark. Accordingly, the dataset consists of 1,000 conversations with an average of 7.36 turns per dialogue, covering five different domains: attraction, hotel, restaurant, taxi, and train.

To examine the characteristics of multi-user interactions, we analyze the distribution of speech acts of $U_t^{(2)}$ within the dataset, which is depicted in Figure 3. Among all user$_2$ utterances, 44% are constatives, 8% are directives, 20% are commissives, 12% are acknowledgments, and 16% are none (i.e., no $U_t^{(2)}$ at t).

Next, we compare the length of $U_t^{(2)}$ and $U_t^{(1)}$ in our dataset. On average, original utterances of $U_t^{(1)}$ contain 11.73 words, while the newly generated $U_t^{(2)}$ are slightly shorter, averaging 10.26 words. This difference arises because $U_t^{(2)}$ often consists of clarifications or brief confirmations rather than full-length requests or responses. M_t tends to be longer, averaging 16.05 words, as they provide detailed responses or requests for clarification.

B Implementation Details

This section outlines implementation details to reproduce the experimental findings of our study. First, we used GPT-4o-2024-08-06 as $\mathcal{L}$ to extend the MultiWOZ 2.1 dataset, following the procedure stated in Section 2.2. Second, for the experiment in Section 3, we used GPT-4o-2024-08-06, claude-3-5-sonnet-20241022, gemini-2.0-flash-001, Meta-Llama-3-8B-Instruct, and gemma-2-9b-it for each model. For hyperparameters such as temperature and top-p, we followed the default setup of each model.

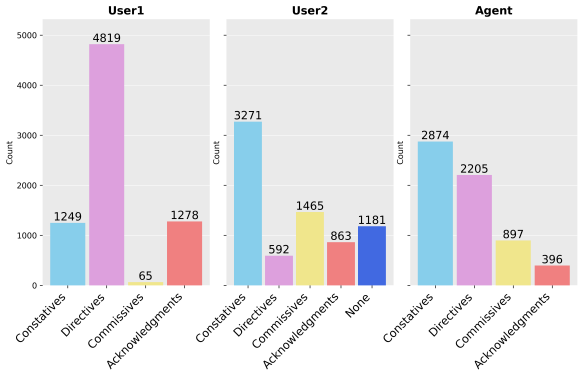

Figure 3: Distribution of speech acts of user$_1$, user$_2$, and agent. We used our speech act type identification method stated in Section 2.2 to classify the speech act type of user$_1$ and Agent for comparison with user$_2$.

Models	Attr.	Hotel	Rest.	Taxi	Train	Avg.
GPT-4o (Zero-shot)	56.8	46.0	55.1	69.3	61.9	57.82
w/ user$_2$ utterances	-4.6	-2.1	-2.2	-1.7	-7.1	-3.54
GPT-4o (One-shot)	57.2	45.6	54.9	68.2	66.2	58.42
w/ user$_2$ utterances	-7.0	-0.7	-2.1	-0.3	-0.8	-2.16

Table 2: Performance comparison between zero-shot and one-shot prompting using GPT-4o. Please refer to 3 for slot-level metrics.

C Additional Experiment with One-shot Example

In this section, we explore the effectiveness of one-shot prompting as a strategy to mitigate the performance degradation of multi-user DST. For this, we randomly selected a dialogue from the MultiWOZ 2.1 training set and extended it with user$_2$ utterances using our methodology to serve as an example in the prompt. The results in Table 2 indicate that, even when providing a multi-user DST example in the prompt, the same degradation trend with user$_2$ utterances persists. These findings further support our observation that multi-user DST remains a challenge for state-of-the-art LLMs.

D Slot-Level Evaluation

In this section, we present the experimental results measured by slot-level accuracy metrics to supplement Section 3.2, which used JGA as an evaluation metric (Dey et al., 2022; Wang and Xin, 2022; Yang et al., 2023; Luo et al., 2024). In this evaluation, we used the following metrics:

- **Accuracy — "None"**: Accuracy when the ground truth is "None" and the model correctly predicts "None".

- **Accuracy — "Dontcare"**: Accuracy when the user indicates no preference ("dontcare") and the model correctly captures it.
- **Accuracy — "Copy_value"**: Accuracy when the model correctly copies a value from the user's utterance.
- **Accuracy — "True"**: Accuracy when the ground truth is "true" and the model predicts "true".
- **Accuracy — "False"**: Accuracy when the ground truth is "false" and the model predicts "false".
- **Accuracy — "Refer"**: Accuracy in correctly identifying references to previously mentioned slot values.
- **Accuracy — "Inform"**: Accuracy in correctly predicting values explicitly provided (informed) by the user.
- **Slot Accuracy (SA)**: Slot Accuracy evaluates the model's ability to correctly predict each individual (domain, slot, value) triplet at every dialogue turn. Unlike JGA, which requires the entire set of predicted belief states to exactly match the ground truth, SA compares each slot prediction independently. A prediction is counted as correct only if the domain, slot name, and slot value all match the ground truth (Wu et al., 2019).

The results of this analysis are presented in Table 3. They demonstrate that our findings in JGA are consistent with the slot-level metrics, which provide a more fine-grained evaluation compared to JGA. These metrics also reveal the same trend discussed in Section 3.2: models such as GPT-4o and Claude 3.5 Sonnet experience more pronounced performance degradation, further supporting our observations.

E Verification of Data Quality

E.1 Human Evaluation

In this paper, we introduced a novel method for incorporating a second user's utterance to simulate a multi-user dialogue environment. This section outlines the manual evaluation conducted to assess the quality of the generated utterances and the dataset constructed using our approach.

To perform this evaluation, we manually reviewed 100 dialogues—representing 10% of the total dataset—based on specific criteria. Three evaluators, all volunteers from the authors' research group, were tasked with assessing the selected dialogues according to the following dimensions:

- **Absence of Bias**: Identification of biased or inappropriate content in user_2 utterances.
- **Utterance Quality**: Assessment of fluency and coherence within the dialogue context.
- **Slot Consistency**: Verification that introducing user_2 does not inadvertently alter slot values, which would introduce noise.

The results of this evaluation are summarized in Table 4. First, no instances of biased or inappropriate content were found in the reviewed dialogues, confirming that user_2 utterances consistently maintained a neutral and respectful tone. We attribute this to the controlled generation process, where user_2 responses are tightly constrained by the context provided by user_1.

Second, 92.68% of user_2 utterances were judged as natural and contextually appropriate, indicating that our generation method effectively preserves dialogue coherence. The remaining cases exhibited minor awkward phrasing or slight tonal mismatches, but these did not significantly hinder comprehension.

Third, slot value consistency was maintained in 94.51% of the dialogues, suggesting that the inclusion of user_2 rarely introduced unintended semantic shifts. In the few inconsistent cases, the discrepancies were due to subtle rephrasings that altered the slot interpretation slightly, rather than clear-cut errors. Notably, this consistency rate reflects a deliberately strict evaluation policy: evaluators were instructed to mark a slot as inconsistent whenever ambiguity was present, regardless of dataset constraints.

For example, consider the following exchange:

- **User1**: "I need to find a hotel that has free parking."
- **User2**: "Make sure the hotel offers free parking for both of us."
- **Slot**: hotel-parking = Yes

Models	"None"	"Dontcare"	"Copy_value"	"True"	"False"	"Refer"	"Inform"	SA	JGA
GPT-4o (Zero-shot)	94.8	74.6	90.6	96.6	89.7	7.8	64.0	94.4	57.82
w/ $user_2$ utterances	-3.4	-29.3	-40.6	-27.8	-22.8	-0.1	-25.8	-5.2	-3.54
GPT-4o (One-shot)	94.7	83.2	92.2	67.3	89.0	9.7	68.3	94.3	58.42
w/ $user_2$ utterances	-0.1	-4.7	-0.8	-2.9	-1.4	-1.9	-4.0	-2.0	-2.16
o4-mini	94.6	80.2	92.7	67.1	88.3	6.5	59.2	94.1	56.82
w/ $user_2$ utterances	-0.1	-0.9	-0.8	-0.3	-1.4	-0.3	-1.4	-0.6	-2.61
Claude 3.5 Sonnet	96.2	75.9	93.6	66.8	87.6	9.7	63.3	95.7	62.08
w/ $user_2$ utterances	-0.2	-2.2	-0.3	-0.7	-10.7	-1.3	-0.8	-3.3	-3.26
Gemini-2.0-Flash	88.2	4.7	20.7	26.7	23.4	0.8	19.2	84.8	35.26
w/ $user_2$ utterances	-0.2	-0.2	-0.3	-2.7	-0.6	-0.3	-0.3	-0.2	-2.32
LLaMA-3.1-8B	86.0	27.6	29.1	39.4	29.0	5.9	23.4	83.1	27.96
w/ $user_2$ utterances	-0.8	-17.3	-8.1	-12.2	-6.9	-0.1	-7.0	-1.1	-1.06
Gemma-2-9B	91.3	34.5	74.8	60.8	57.2	15.4	52.8	90.2	43.52
w/ $user_2$ utterances	-0.7	-1.7	-2.4	-0.6	-2.8	-2.5	-2.1	-0.8	-1.36

Table 3: The performance of each LLM model for zero-shot DST in slot-level metrics.

	Absence of Bias	Utterance Quality	Slot Consistency
Ratio (%)	100.00	92.68	94.51

Table 4: Results of the human evaluation. A higher ratio indicates better performance.

	Attr.	Hotel	Rest.	Taxi	Train	Avg.
GPT-4o	39.4	43.4	60.9	62.8	62.6	53.82
w/ $user_2$ utterances	-7.4	-2.6	-2.2	-0.5	-5.3	-3.60

Table 5: Performance of GPT-4o on clean dialogues without inconsistent slots.

In this case, the slot was marked as inconsistent, based on the interpretation that $user_2$ implicitly requests an additional parking space. However, in MultiWOZ 2.1, the `hotel-parking` slot is binary (e.g., `yes`/`no`), and the model is not expected to distinguish between single or multiple parking requests. Thus, while human annotators may perceive this as inconsistent, such differences are typically irrelevant to the DST model. In this sense, the reported 5.49% inconsistency rate should be viewed as an upper bound, reflecting a conservative human-centric judgment rather than an estimate of likely performance degradation (e.g., in JGA).

E.2 Ablation Study on Clean Dialogues

In addition to the human evaluation, we conducted an ablation study using 100 dialogues that had been flagged by evaluators as containing no slot inconsistencies. This experiment was carried out using GPT-4o, and the results are presented in Table 5. The findings reveal a similar pattern of performance degradation, even in these clean dialogues. This suggests that the minor inconsistencies identified during manual inspection do not significantly affect the overall conclusions of the study.

F Error Analysis

This section provides an analysis of the errors induced by the introduction of $user_2$ utterances in multi-user DST. We manually investigated cases where the LLM correctly inferred slot values in the original dataset but misclassified them when the dataset was extended with additional user utterances. Errors primarily fall into two categories: (1) errors in extracting the correct slot value and (2) errors in identifying the correct slot type. Below, we provide examples and discussions for each type.

Table 6 shows a conversation where the user asks for information about "Warkworth House." In the single-user scenario, the LLM correctly identifies "Warkworth House" as the value of the *Hotel-Name* slot. However, once an additional $user_2$ utterance is introduced (expressing curiosity about the history of Warkworth House), the model fails to track the intended hotel name. This suggests that the extra mention of historical information deflects the model's attention, causing it to overlook the slot value relevant to $user_1$'s primary request. Importantly, the problem seems to stem from the model's difficulty in separating the primary user request (i.e., $user_1$ is explicitly asking for hotel details) from the ancillary utterance by $user_2$. Even

though the instructions clearly state that $user_1$'s state should be tracked, the presence of another speaker's query creates an implicit context shift. As a result, the model may conflate or ignore the critical slot values it had previously captured.

In addition to missing values, multi-user input can also cause the model to produce erroneous slot types. Table 7 demonstrates an example where the user requests a Mediterranean restaurant. In the single-user context, the LLM assigns the *Restaurant-Food* slot as "Mediterranean," which is correct. However, once $user_2$ joins the conversation expressing enthusiasm for a "Mediterranean experience" in the center of town, the model additionally (and incorrectly) infers a *Restaurant-Area* slot value of "dontcare." This misprediction underscores how $user_2$'s mention of a location can trigger confusion about the user's overall constraints or preferences. The second user's statements appear to "override" or complicate the original request, leading the model to infer a new slot type (*Restaurant-Area*) even though $user_1$ never expressed indifference about the area. Such discrepancies highlight a lack of robust multi-user grounding in the model: it fails to distinguish separate $user_1$ requirements from $user_2$ remarks.

These errors collectively illustrate the challenges in scaling dialogue state tracking to multi-user scenarios. Both examples suggest that LLMs struggle to track the request of the primary user and disentangle the utterance of the other user. In multi-user dialogue applications (e.g., group travel planning or collaborative booking platforms), these behaviors could significantly degrade user experience and system reliability. Future efforts must address how best to isolate or merge multiple users' goals to avoid contaminating one user's state with another's. Potential mitigation strategies include more explicit prompts that separate $user_1$'s goals from $user_2$'s utterance, or neural architectures that model each speaker's state in parallel and then reconcile them at the system level. Overall, our results confirm that incorporating additional speakers in dialogue amplifies existing challenges in DST, highlighting the need for further research into robust, speaker-aware state tracking mechanisms.

Original MultiWOZ 2.1 w/o $User_2$ Utterance
User: Do you have information about the Warkworth House? *[Hotel-Name: Warkworth House]*
Extended MultiWOZ 2.1 w/ $User_2$ Utterance
User1: Do you have information about the Warkworth House? **User2**: I'm curious about the history of Warkworth House too! *[Hotel-Name: ?]*

Table 6: The error case where the LLM failed to extract the correct slot value when the $user_2$ utterance is introduced.

Original MultiWOZ 2.1 w/o $User_2$ Utterance
Agent: I have curry garden for Indian in the centre of town, but no south indian. **User**: What about one that serves mediterranean? *[Restaurant-Food: mediterranean]*
Extended MultiWOZ 2.1 w/ $User_2$ Utterance
Agent: I have curry garden for Indian in the centre of town, but no south indian. **User1**: What about one that serves mediterranean? **User2**: I'm in for a Mediterranean experience. Let's explore some top-notch options in the center together! *[Restaurant-Food: mediterranean, Restaurant-Area: dontcare]*

Table 7: The error case where the LLM failed to predict the correct slot type when the $user_2$ utterance is introduced.

G Prompt Design

G.1 Prompt for Identification of Speech Act Type

Note that we shuffle the order of four options for each turn, to avoid biased identification (Lu et al., 2022).

```
Examine the previous dialogue history and choose the appropriate strategy for the utterance of User2 among the four options
    ↪ below. Considering the previous conversation, choose appropriate, unbiased representation for diversity.

Dialogue history so far:
{DIALOGUE HISTORY}

Current turn:
Agent: {AGENT UTTERANCE}
User1: {USER1 UTTERANCE}
User2: <New utterance to be generated>

Your task and requirements:
- Select one option among the four options mentioned below:
    - Constatives : committing the speaker to something's being the case (answering, claiming, confirming, denying,
        ↪ disagreeing, stating)
    - Directives : attempts by the speaker to get the addressee to do something (apologizing, greeting, thanking, accepting an
        ↪  acknowledgment)
    - Commissives : committing the speaker to some future course of action (advising, asking, forbidding, inviting, ordering,
        ↪ requesting)
    - Acknowledgments : express the speaker's attitude regarding the hearer with respect to social action (promising, planning
        ↪ , vowing, betting, opposing)
- Output must be same words in prompt type option "Constatives", "Directives", "Commissives", "Acknowledgments".
```

G.2 Prompt for Utterance Generation

```
You are User2, a friend of User1. Your role is to casually communicate with User1 between Agent in less than 20 words
    ↪ following generation guide. User1 and User2 are talking to Agent.

Dialogue history so far:
{DIALOGUE HISTORY}

Current turn:
User1: {USER1 UTTERANCE}
User2: <New utterance to be generated>
Fixed slots: {SLOT VALUE OF CURRENT TURN}

Your task and requirements:

- Generate an appropriate utterance for User2 to respond to User1.
- Do not modify existing Current fixed slot and values or introduce new slot and values.
- Do not generate repeated utterances. (Ex. starting with 'Let's'or 'That sounds')
- Do not ask questions to User1. Only ask to Agent. Because the Agent will respond in the next turn.\n"
- Do not mention explitcitly word 'Agent'.\n"
- Note that you are in same circumstances with User1.
- Types of generated dialogues of User2: {UTTERANCE TYPE}
```

G.3 Prompt for Validation of Generated Utterance

```
You will be given a conversation sequence consisting of three parts:
1. Agent's response to previous Generated_User1 (`A`): {AGENT UTTERANCE}
2. User1's original utterance (`U1`): {USER1 UTTERANCE}
3. Generated_User2's transformed utterance (`U2`): {USER2 UTTERANCE}

Your task is to check the consistency of the sequence:
1. Check if User2's utterance (`U2`) is logically consistent with User1's original utterance (`U1`).
2. Verify that Generated_User2's (`U2`) slots match the current slot values from User1 (`U1`) slot: {SLOT VALUE OF CURRENT
    ↪ TURN}.
Output 'True' if both criteria 1 and 2 are met. Otherwise, output 'False'. Output: [True / False]
```

G.4 Prompt for Experiment

Note that this is a slightly updated version of the prompt design suggested by the previous study, considering the existence of $user_2$ (Heck et al., 2023).

```
Consider the following list of concepts, called "slots" provided to you as a json list.

"slots": {
    "taxi-leaveAt": "the departure time of the taxi",
    "taxi-destination": "the destination of the taxi",
    "taxi-departure": "the departure of the taxi",
    "taxi-arriveBy": "the arrival time of the taxi",
    "restaurant-book_people": "the amount of people to book the restaurant for",
    "restaurant-book_day": "the day for which to book the restaurant",
    "restaurant-book_time": "the time for which to book the restaurant",
    "restaurant-food": "the food type of the restaurant",
    "restaurant-pricerange": "the price range of the restaurant",
    "restaurant-name": "the name of the restaurant",
    "restaurant-area": "the location of the restaurant",
    "hotel-book_people": "the amount of people to book the hotel for",
    "hotel-book_day": "the day for which to book the hotel",
    "hotel-book_stay": "the amount of nights to book the hotel for",
    "hotel-name": "the name of the hotel",
    "hotel-area": "the location of the hotel",
    "hotel-parking": "does the hotel have parking",
    "hotel-pricerange": "the price range of the hotel",
    "hotel-stars": "the star rating of the hotel",
    "hotel-internet": "does the hotel have internet",
    "hotel-type": "the type of the hotel",
    "attraction-type": "the type of the attraction",
    "attraction-name": "the name of the attraction",
    "attraction-area": "the area of the attraction",
    "train-book_people": "the amount of people to book the train for",
    "train-leaveAt": "the departure time of the train",
    "train-destination": "the destination of the train",
    "train-day": "the day for which to book the train",
    "train-arriveBy": "the arrival time of the train",
    "train-departure": "the departure of the train"
}

Some "slots" can only take a value from predefined list:

"categorical": {
    "hotel-pricerange": ["cheap", "moderate", "expensive"],
    "hotel-area": ["north", "south", "east", "west", "centre"],
    "hotel-parking": ["yes", "no"],
    "hotel-internet": ["yes", "no"],
    "hotel-type": ["hotel", "guest house"],
    "restaurant-pricerange": ["cheap", "moderate", "expensive"],
    "restaurant-area": ["north", "south", "east", "west", "centre"],
    "attraction-area": ["north", "south", "east", "west", "centre"]
}

Now consider the following dialogue between more than two parties called the "system", "user1", and "user2".

Can you make answer as JSON ouput format which of the "slots" were updated by the "user1" in its latest response to the "
    ↪ system" refer to "slots" JSON list?

Present the updates in JSON format for each "slot" that was updated. There has no more than one update per "slot".
If no "slots" were updated, return an empty JSON list. If you encounter "slots" that were requested by the "user1" then fill
    ↪ them with "?". If a user does not seem to care about a discussed "slot" fill it with "dontcare".
```