Dialog Flow Induction for Constrainable LLM-Based Chatbots

Stuti Agrawal*, **Pranav Pillai***, **Nishi Uppuluri***, **Revanth Gangi Reddy, Zoey Li,**
Gokhan Tur, Dilek Hakkani-Tur, Heng Ji
University of Illinois Urbana-Champaign
{stutia3, nu4, ppillai3, revanth3, shal2}@illinois.edu
{gokhan, dilek, jih}@illinois.edu

Abstract

LLM-driven dialog systems are used in a diverse set of applications, ranging from healthcare to customer service. However, given their generalization capability, it is difficult to ensure that these chatbots stay within the boundaries of the specialized domains, potentially resulting in inaccurate information and irrelevant responses. This paper introduces an unsupervised approach for automatically inducing domain-specific dialog flows that can be used to constrain LLM-based chatbots. We introduce two variants of dialog flow based on the availability of in-domain conversation instances. Through human and automatic evaluation over various dialog domains, we demonstrate that our high-quality data-guided dialog flows[1] achieve better domain coverage, thereby overcoming the need for extensive manual crafting of such flows.

1 Introduction

The widespread use of Large Language Models (LLMs) (OpenAI et al., 2023) for chatbots, highlighted by their human-like conversational abilities across many topics, faces challenges in specialized domains due to their tendency to go off-topic. This generalization capability, while a strength, necessitates the development of more effective control mechanisms to ensure chatbots remain within the desired domain of conversation, especially in specialized fields such as healthcare or legal advice. Controlling LLM-based chatbots can be effectively managed through dialog flows or schemas[2] (Bohus and Rudnicky, 2009; Mosig et al., 2020), which structure conversations along predefined paths of dialog actions, acting as directed graphs where nodes represent actions by the user or bot, and edges are the transitions between actions. This structure helps steer the conversation, keeping it within relevant topics, and also enables chatbots to adapt to new tasks or domains without prior training (Zhao et al., 2023).

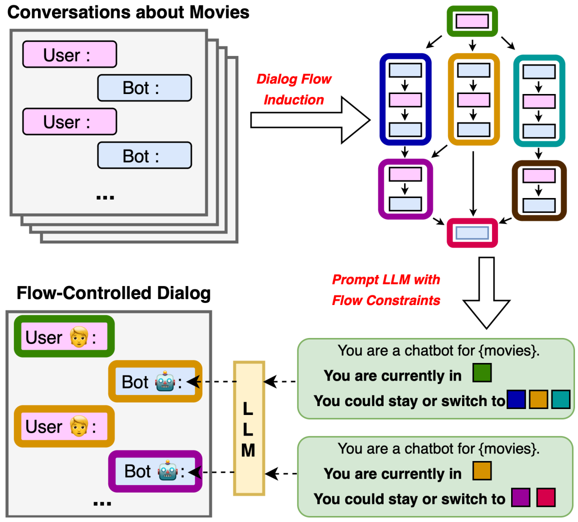


Figure 1: Figure demonstrating how automatically induced domain-specific dialog flows can be used to constrain chatbots to produce domain-focused responses.

However, the construction of precise dialog flows is challenging (Huang et al., 2020), given the diversity of dialog in different domains. The most prevalent approaches (Mehri and Eskenazi, 2021; Zhao et al., 2023) use schemas that are carefully handcrafted by the dialog system developers. The design of dialog schemas thus has significant manual overhead for developers, resulting in scalability and coverage limitations (Zhang et al., 2020).

This paper introduces an unsupervised method to generate domain-specific dialog flows, exploiting GPT-4's knowledge to systematically create detailed dialog flows reflecting conversational patterns in various domains. We begin by prompting GPT-4 to produce a structured representation of dialog interactions between users and bots, and then further refine this through self-reflective feedback

*denotes equal contribution
[1]Code is available at https://github.com/gangiswag/dialog-flows
[2]We use the terms *flows* and *schemas* interchangeably. Our definition of dialog schemas follows Mosig et al. (2020) to be analogous to task specifications, different from task slots.

Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 66–77
September 18–20, 2024. ©2024 Association for Computational Linguistics

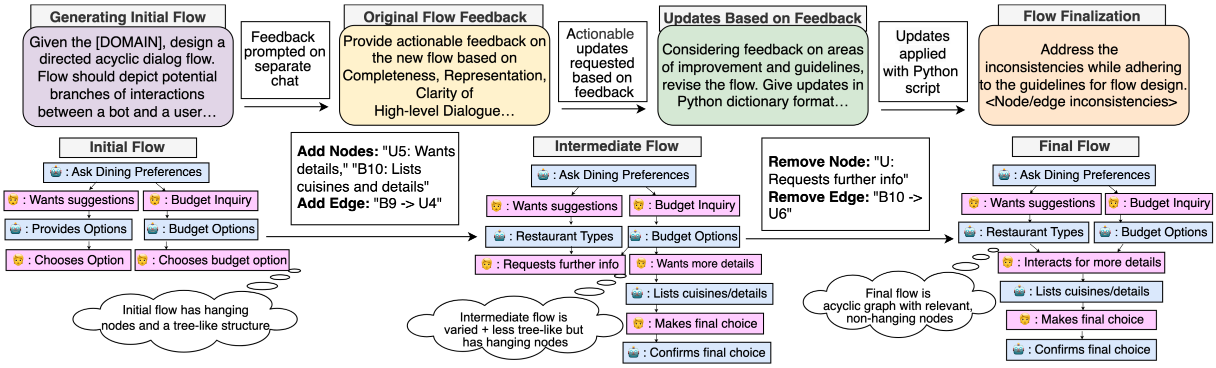


Figure 2: Figure showing the process for intrinsic flow induction. An initial flow is first generation which is further refined with feedback, update, and clean-up stages. Detailed prompts for each stage are provided in the appendix.

based on a set of predefined criteria (see figure 2).

Further, when we have domain-specific conversations, our approach automatically identifies distinct user and bot dialog actions within these conversations (see figure 3). These dialog actions, along with selected conversations that exemplify each action, are used to condition the GPT-4 prompt to ensure the dialog flows are grounded using actual domain instances. This approach enables the automated creation of structured dialog flows, facilitating the development of effective domain-specific chatbots that adhere to their domain's conversational boundaries. Our main contributions are:

- This paper introduces an approach for automatically constructing dialog flows for various domains in an unsupervised manner.
- The proposed method uses a multi-step framework, that can further leverage domain-specific dialog instances, leading to a graph-like flow illustrating the structure of conversations in the domain.

2 Dialog Flow Induction

A dialog flow is a flowchart comprising nodes which can be a user or bot dialog action, and edges that denote logical flow or transitions between these actions. Dialog flows are tailored to different domains. Figure 2 shows an excerpt of a dialog flow, with more detailed examples in the appendix. In this section, we detail our approach for automatically inducing the dialog flow for a given conversation domain. Specifically, we induce two variants of dialog flows, namely *intrinsic flows* (in §2.1) or *data-guided flows* (in §2.2) depending on whether sample conversations in the domain are available.

2.1 Intrinsic Dialog Flow

When domain-specific conversation data is unavailable, we propose to induce dialog flows using the *intrinsic* domain-related knowledge of LLMs and their understanding of conversational principles. Our intrinsic flow induction process starts with GPT-4 creating an initial flow based on the domain's name. Next, GPT-4 self-evaluates the flow based on predetermined guidelines, to provide concrete actionable feedback for improvement. Using this feedback, GPT-4 then suggests a set of edits, which are automatically applied to the initial flow. Finally, automated checks are run to identify inconsistencies in the flow, which GPT-4 then handles in the end clean-up stage. Figure 2 shows the overall intrinsic flow induction process, with more details on each step provided below.

Initial Flow Generation: The flow induction starts with prompting GPT-4 with a specific generation prompt to create a dialog flow, as shown in Figure 2. Along with the domain name, the prompt includes details on the intended structure of the dialog flow. After the initial flow is generated, it undergoes further refinement as detailed next.

Flow Feedback and Updates: The initial flow often suffers from low coverage along with ambiguous or repetitive action labels for bot and user nodes. We address these by leveraging GPT-4 for self-assessment (Bai et al., 2022) and refining the dialog flow based on the feedback. The refinement process starts by obtaining GPT-4 feedback based on the following aspects:

- **Representativeness:** Both the bot and user actions should be relevant to the domain, and should not be vague or generic.

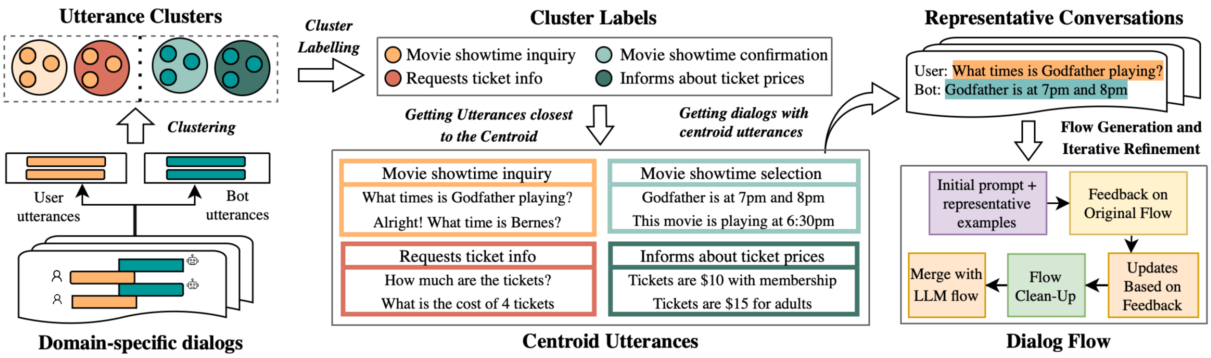


Figure 3: Figure showing the methodology for inducing dialog flows using a data-guided approach. Representative examples from the domain conversation instances are used to condition the GPT-4 prompts.

- **Coverage:** Ensuring the flow captures a broad range of conversational possibilities relevant to the domain.
- **Clarity of Dialog Action:** Each node should reflect a clear and meaningful dialog action.
- **Optimality:** Eliminate redundancy, ensuring no nodes depict overlapping dialog actions.

Based on the shortcomings identified by the self-reflective feedback, GPT-4 is then prompted to output a set of concrete updates to be made to the flow, which can include nodes or edges to add, remove, or edit. To control for the extent to which the flow changes, the updates are performed with an automated Python script rather than directly prompting GPT-4 to apply the updates[3].

Flow Finalization: Finally, the dialog flow undergoes a clean-up stage where trivial inconsistencies, such as dangling non-terminal nodes, bot-bot or user-user connections, are identified. These are passed as input to GPT-4 along with a final prompt, to ensure the flow is structurally correct.

2.2 Data-Guided Dialog Flow

The intrinsic dialog flow induction approach, while expansive in its scope, relies predominantly on the model's inherent knowledge of the typical interactions and transitions that could occur within the specified conversation domain. However, when dialog instances within the given domain are provided, the intrinsic flow can be updated to include actual conversational patterns. We call this approach *data-guided* flow induction, which aims to mirror real-world dialog dynamics. Specifically, the approach conditions the GPT-4 flow generation prompt with representative examples in the form of action labels and sample conversations for the domain, which help ground the flow to real-life conversation data. Figure 3 gives an overview of data-guided flow induction process, with more details provided below.

Identifying Representative Examples: Given dialog instances for a domain, the following steps identify the user and bot actions, along with sample conversations that are representative of the domain.

- ***Clustering and Labeling***: The user and bot utterances from dialogs in the domain are clustered separately using SentenceBert (Reimers and Gurevych, 2019) embeddings. Next, GPT-4 is prompted to label each cluster with a dialog action by providing it with the utterances closest to each centroid.
- ***Cluster Merging***: Next, we merge clusters that exhibit significant overlaps in terms of action intent, based on the cosine similarity between the labels. This reduces the redundancy in the action labels by grouping clusters with similar actions.
- ***Picking sample conversations***: Finally, the conversations that include utterances corresponding to the cluster centroids are picked as the representative dialog instances to include in the GPT-4 prompt for flow generation. This ensures that the conversations encompass a wide spectrum of dialog actions and user intents specific to the domain.

Flow Generation: As shown in Figure 3, the flow induction follows a similar generation process as the intrinsic dialog flow. Firstly, the representative action labels and sample conversations for the domain are included in the initial flow generation prompt. Next, the feedback, update, and clean-up steps are applied to result in a dialog flow.

[3]We hypothesize that this provides the ability to heuristically control different aspects of the dialog flow, such as depth, breath, density of edges, etc.

Merging with Intrinsic Flow: The intrinsic flow approach creates broad, expansive dialog flows, but can still fall short of reflecting domain-specific patterns from real-world conversations. On the other hand, solely relying on the domain dialog instances can hurt extensiveness, as they can have limited variability. Hence, we adopt a hybrid approach for the data-guided flow by merging the intrinsic flow with the flow induced solely from domain-specific data. This capitalizes on the extensive scope of the intrinsic flow with the detailed focus from domain data. This merging step is achieved by prompting GPT-4 to identify and retain distinctive features from the intrinsic flow, while removing redundant elements. We call this final flow, *data-guided* flow.

3 Experiments

We perform both human and automatic evaluations to assess the induced dialog flows.

3.1 Datasets

Open-domain dialog can involve a single conversation touching upon different domains, such as movies, sports, music, etc. Hence, for simplicity, we consider domains from task-oriented dialog in our experimental settings, wherein the domains are distinct and correspond to the end user task, such as movie tickets, flight booking, restaurant reservations, etc. We consider a dialogs across various task-oriented domains, comprising 24 domains[4] from MetaLWoz (Shalyminov et al., 2019) and 5 domains from MultiWOZ (Budzianowski et al., 2018). For the data-guided flow induction, for each domain, we utilized 80% of the data as domain-specific instances available for training, with the remaining 20% reserved for evaluating coverage of the bot-bot transitions (described later in §3.3).

3.2 Human Evaluation of Flow Quality

The evaluators (five undergraduate computer science students) were tasked with examining data-guided and intrinsic flows across the 24 different domains from MetaLwoz. The evaluators were given detailed guidelines (provided in the appendix), and were instructed to assess each flow on a scale of 1 to 5 for *domain coverage*, *conclusiveness* and *coherence*.

Table 1 shows numbers from human evaluation of the data-driven and intrinsic dialog flows. The numbers (expanded to a scale of 20-100) are averaged over all the domains, with flows for each domain being annotated by 5 evaluators. We can see that the data-driven flow, on account of leveraging domain-specific dialog instances, improves over the intrinsic flow on domain coverage. Further, both dialog flows have similarly high scores for conclusiveness and coherence, implying our unsupervised approach, by leveraging GPT-4, can automatically induce high-quality dialog flows. We employed Randolph's kappa to evaluate the multi-rater agreement. Our findings revealed a kappa value of 0.32, indicating a fair level of agreement across the board. Specifically, the domain coverage metric exhibited the highest kappa value of 0.46, signifying moderate agreement.

[4]We excluded domains that had ambiguous or generic names, such as Play Times, Catalogue, Agreement Bot, etc.

	Intrinsic	Data-driven
Domain Coverage	90.7	**93.0**
Conclusiveness	**87.8**	87.7
Coherence	84.5	**84.8**

Table 1: Results from human evaluation (in %) of different aspects of the induced dialog flows

Dataset	Intrinsic	Data-driven
MetaLWoz	31.6	**33.1**
MultiWOZ	39.9	**43.0**

Table 2: Bot-Bot transition coverage (in %) for the proposed variants of dialog flows on the MetalWoz (Shalyminov et al., 2019) and MultiWOZ (Budzianowski et al., 2018) datasets. Detailed domain-wise numbers are provided in Table 3 in the appendix.

3.3 Automatic Evaluation of Flow Coverage

Next, we automatically evaluated the domain coverage of different dialog flows, by measuring the coverage on capturing bot-to-bot transitions within the domain conversations in the test set. We leveraged Mistral-7B-Instruct (Jiang et al., 2023) to classify bot utterances into the most appropriate node in the dialog flow. We then examined whether the next bot utterance mapped to the directly succeeding node in the dialog flow. Essentially, this metric measures the percentage of bot-bot transitions in domain conversations that conform to the given dialog flow. Table 2 shows numbers for automatic coverage evaluation. We can see that the data-driven dialog flow has better coverage of the domain's bot-bot transitions.

4 Conclusion and Future Work

We introduce a novel method for developing dialog flows that reflect the combined intrinsic knowledge of LLMs and existing domain-relevant dialogs. Our data-driven dialog flow approach achieves better domain coverage than the intrinsic flow approach across human and automatic evaluations. Our paper outlines a blueprint (in Figure 1) for integrating the generated dialog flows into LLM-based chatbots, with a primary focus on the methodologies for dialog flow generation. We believe these dialog flows can be a springboard for future interactive dialog systems that maintain a natural conversation flow within the domain.

Limitations

In this study, our experimentation was confined to task-oriented dialogs, encompassing a relatively narrow spectrum of dialog flows. This specialization may limit the applicability of our findings to dialog domains characterized by a broader array of tasks and more open-ended dialogues. Additionally, our methodology relies solely on unsupervised clustering techniques, bypassing datasets that are annotated with slot values and user intents, which could potentially enhance dialog flow induction. Furthermore, we have not extended our research to test the performance of chatbots constrained by the dialog schemas we developed. Therefore, the efficacy of these schemas in practical chatbot applications remains an area for future investigation.

Acknowledgment

We would like to thank the CS STARS program at UIUC for supporting Stuti and Nishi. We are grateful to members of the BlenderNLP group for their valuable comments and feedback. This research is based on work supported by U.S. DARPA KAIROS Program No. FA8750-19-2-1004 and U.S. DARPA INCAS Program No. HR001121C0165. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Dan Bohus and Alexander I Rudnicky. 2009. The ravenclaw dialog management framework: Architecture and systems. *Computer Speech & Language*, 23(3):332–361.

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 the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.

Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2020. Challenges in building intelligent open-domain dialog systems. *ACM Transactions on Information Systems (TOIS)*, 38(3):1–32.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Shikib Mehri and Maxine Eskenazi. 2021. Schema-guided paradigm for zero-shot dialog. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 499–508.

Johannes E. M. Mosig, Shikib Mehri, and Thomas Kober. 2020. Star: A schema-guided dialog dataset for transfer learning. *ArXiv*, abs/2010.11853.

OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Igor Shalyminov, Sungjin Lee, Arash Eshghi, and Oliver Lemon. 2019. Few-shot dialogue generation without annotated data: A transfer learning approach. *arXiv preprint arXiv:1908.05854*.

Zheng Zhang, Ryuichi Takanobu, Qi Zhu, MinLie Huang, and XiaoYan Zhu. 2020. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, 63(10):2011–2027.

Jeffrey Zhao, Yuan Cao, Raghav Gupta, Harrison Lee, Abhinav Rastogi, Mingqiu Wang, Hagen Soltau, Izhak Shafran, and Yonghui Wu. 2023. AnyTOD: A programmable task-oriented dialog system. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16189–16204, Singapore. Association for Computational Linguistics.

A Appendix

MetaLWoz	Intrinsic	Data-driven
Alarm set	32.9	**42.2**
Apartment finder	30.9	**45.2**
Bank bot	**34.2**	30.8
Bus schedule	**37.2**	14.4
City info	29.4	**33.4**
Edit playlist	**44.2**	39.4
Event reserve	28.8	**30.5**
Library Request	**35.7**	30.1
Movie listings	30.7	**34.4**
Music suggester	**34.0**	25.3
Name suggester	**43.2**	16.7
Order pizza	31.6	**36.1**
Pet advice	**33.8**	31.7
Phone plan	31.6	**37.8**
Restaurant picker	**29.4**	29.2
Scam lookup	22.6	**31.2**
Shopping	17.0	**22.9**
Ski Bot	27.2	**32.2**
Sports info	36.6	**37.1**
Store details	**35.7**	32.4
Update calendar	**38.4**	28.8
Update contact	**32.5**	30.3
Weather check	**36.1**	29.5
Wedding planner	17.0	**24.2**
Average	31.6	**33.1**

MultiWOZ	Intrinsic	Data-driven
Restaurant	31.0	**43.9**
Hotel	43.2	43.2
Attractions	43.3	**53.3**
Taxi	**75.3**	50.5
Train	6.9	**24.1**
Average	39.9	**43.0**

Table 3: Bot-Bot transition coverage (in %) for the proposed variants of dialog flows when measured on various domains in the MetalWoz (Shalyminov et al., 2019) and MultiWOZ (Budzianowski et al., 2018) datasets.

MetaLWoz	Train	Test
Alarm set	1345	336
Apartment finder	399	100
Bank bot	294	73
Bus schedule	718	180
City info	772	193
Edit playlist	459	115
Event reserve	431	108
Library request	1071	268
Movie listings	486	121
Music suggester	356	89
Name suggester	399	100
Order pizza	462	115
Pet advice	341	85
Phone plan	397	99
Restaurant picker	428	107
Scam lookup	1326	332
Shopping	722	181
Ski bot	486	121
Sports info	449	112
Store details	590	147
Update calendar	1593	398
Update contact	522	131
Weather check	441	110
Wedding planner	408	102

Table 4: Statistics of dialogs in various domains in the MetalWoz (Shalyminov et al., 2019) dataset.

Intrinsic Flow Initial Prompt

Given the context of [DOMAIN], design a directed acyclic dialog flow suitable for visualization with mermaid.js. This flow should depict the nuances and potential branches of interactions between a bot and a user.Please adhere to the following guidelines:

Nodes Definition: Use distinct nodes to represent the bot ("B") and the user ("U").
High-Level Dialog Action: Each node should encapsulate that segment's core sentiment or function in the conversation, relevant to [DOMAIN]. It should be a label for the node representing a high-level dialogue action and not just the dialogue.
Flow & Directionality: Create directed connections between nodes to represent the progression of the conversation. The dialogue should flow from one node to potentially multiple nodes, allowing for various conversational turns.
Diverse Conversational Possibilities: Ensure that bot nodes can lead to multiple user nodes and vice versa. This should account for various user responses or bot prompts, showcasing the range of interactions possible within [DOMAIN].
Acyclic Structure: The dialog flow must not have loops or cyclic pathways. If a similar action or sentiment arises later in the conversation, introduce a new node to represent it, rather than looping back to an earlier node.
Mermaid.js Compatibility: Ensure that the constructed flow is adherent to mermaid.js graph notation, guaranteeing its seamless rendering.

Considering the guidelines, craft a dialogue flow focused on [DOMAIN]. The bot always begins by greeting the user and asking for what they want. The graph should be connected. The bot and user nodes should be in different colors. A bot node is only followed by user nodes and user nodes are by bot nodes.

Intrinsic Flow Feedback Prompt

Based on the below evaluation criteria, suggest some improvements and provide concise + actionable feedback on the flow just generated:

Optimality: Check for redundancy. Ensure that nodes aren't replicating the same or very similar dialog actions, even if they arise at different points in the conversation

Clarity of High-Level Dialog Action: For every node, evaluate if the high-level dialog action is clear and meaningful. Avoid nodes that are vague or overly complex. Can someone unfamiliar with the domain understand the flow and interactions by looking at the flow?

Extensiveness: Does the flow account for diverse conversational possibilities? Are all the nodes interconnected to the graph? Does the flow cover all major high level topics and interactions within the given domain?

Representativeness of the Domain: Bot Nodes (B): Do the bot nodes represent clear and unambiguous actions? Are they too broad or too specific? User Nodes (U): Do user nodes accurately capture an adequate range of potential user responses and inquiries relevant to the domain?

Intrinsic Flow Update Prompt

Taking into consideration the feedback and the original design guidelines - keep it in directed acyclic graph structure and make sure all new components are labeled and connected to the graph correctly- revise the flow. Ensure your revised flow addresses the identified areas of improvement while still adhering to the primary instructions for flow construction. Make sure to account for all new nodes including merged nodes and their labels/colors. Make sure all user nodes connect with bot nodes and bot nodes are the end of the conversations. Give your updates in the below format:

'split_nodes':
'NodeToSplit': ['NewNode1', 'NewNode2', ...],
'add_nodes':
'NodeToAdd': 'Label',
'remove_nodes':
'NodeToRemove1', 'NodeToRemove2', ...

'relabel_nodes':
'NodeToRelabel': 'NewLabel',
'add_edges':
('Start Node', 'End Node'),
'remove_edges':
('Start Node', 'End Node'),

Intrinsic Flow Finalization Prompt

Clean up the flow to create a final flow. Ensure your revised flow addresses the identified areas of improvement while still adhering to the primary instructions for flow construction. Get rid of hanging/loose user nodes (user nodes with no output), have graph in directed acyclic structure, bot nodes shouldn't be connected to other bot nodes, and user nodes shouldn't be connected to other user nodes. All nodes should have input/output except begin and end nodes, one node shouldn't point to the another node more than once, and make sure all bot nodes are correctly colored.

Intrinsic and Data-Guided Flows Merging Prompt

Given the two dialogue flows for the [DOMAIN] bot. One flow is LLM generated and the other is from data examples, merge all the unique elements of the two flows and do not duplicate similar elements. Merge the two flows based on the following design guidelines:
< Design guidelines >

Figure 4: Figure showing prompts for intrinsic and data-guided dialog flow generation.

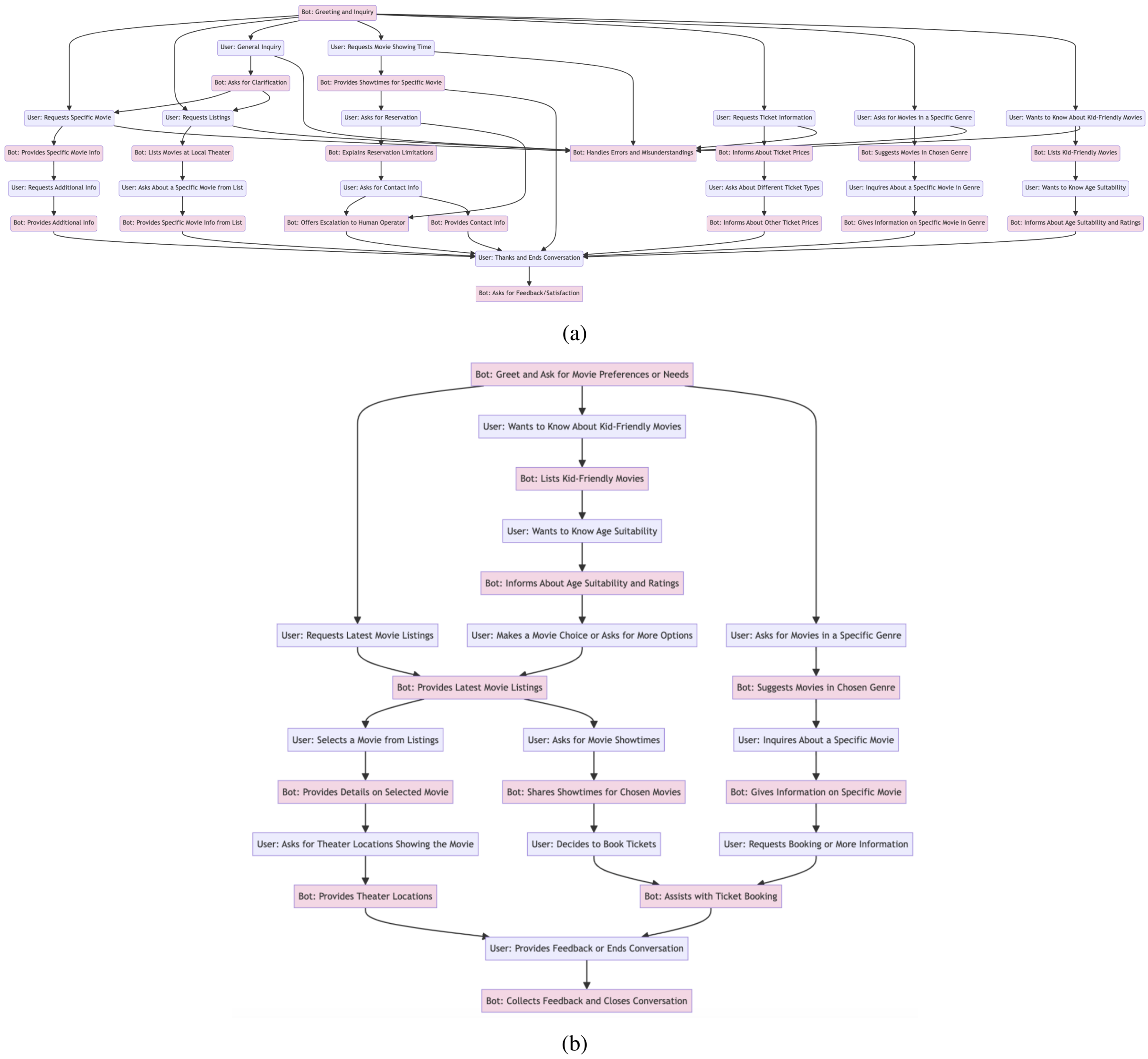
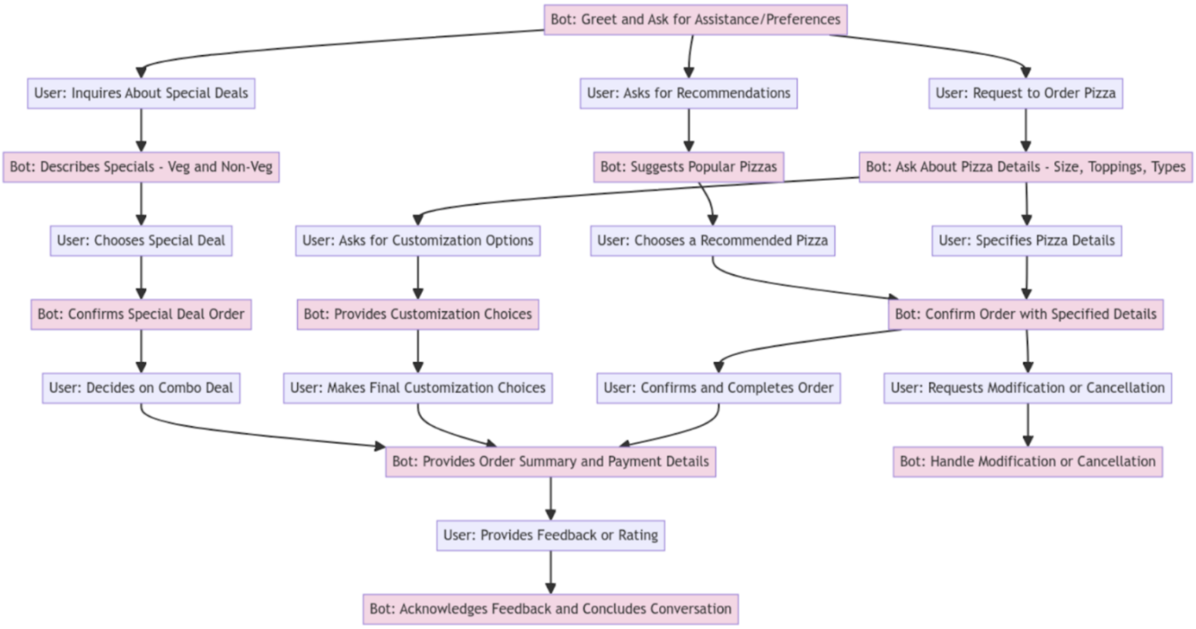
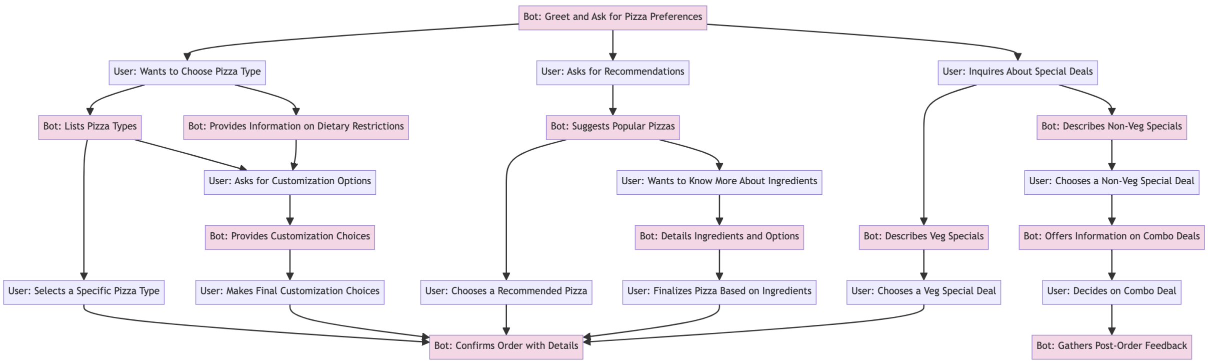


Figure 5: Data-driven (a) and Intrinsic (b) flows for the movie listings domain from MetaLWoz.

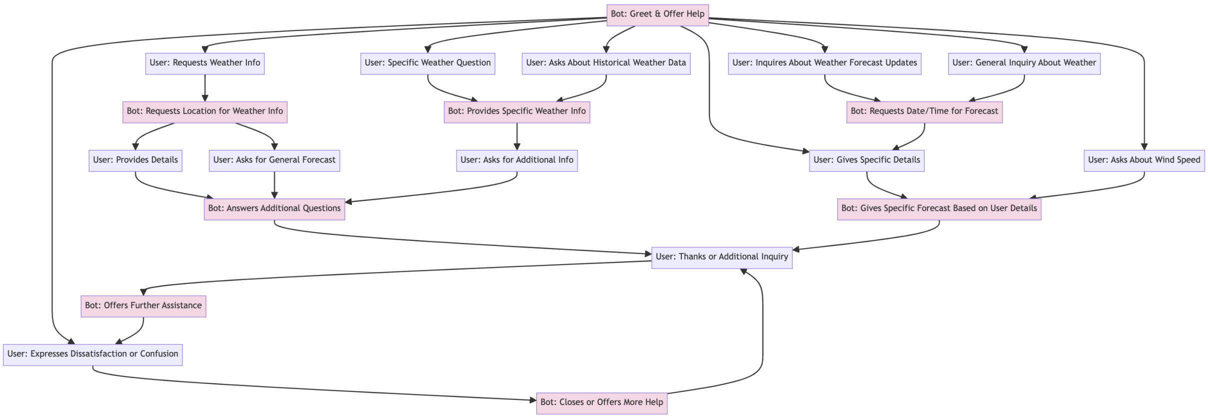


(a)

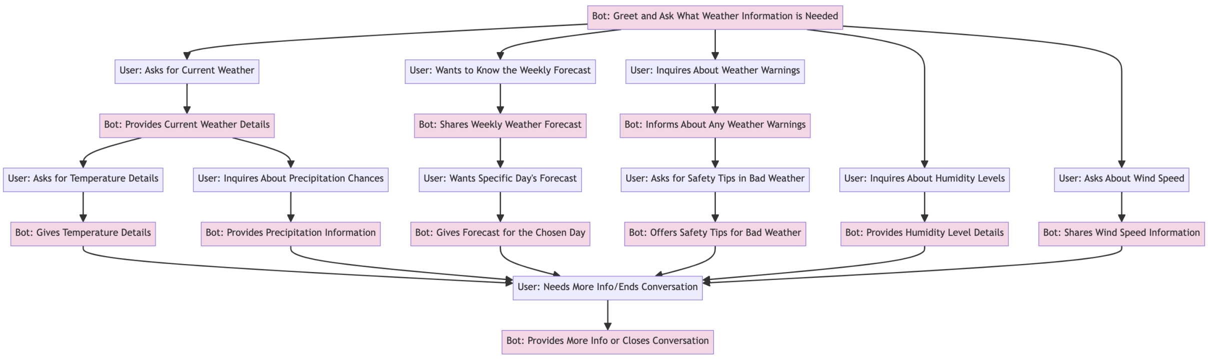


(b)

Figure 6: Data-driven (a) and Intrinsic (b) flows for the order pizza domain from MetaLWoz.



(a)



(b)

Figure 7: Data-driven (a) and Intrinsic (b) flows for the order weather domain from MetaLWoz.

What is a task-oriented dialog flow?
A dialog flow is like a roadmap for conversations between a user and a chatbot, outlining all the possible exchanges they can have. It guides the chatbot on how to respond to different user inputs, ensuring the conversation flows smoothly and logically. Dialog flows are composed of bot nodes, which correspond to bot actions, and user nodes, corresponding to user actions. In the below dialog flows, all bot nodes are shown in pink and all user nodes are shown in blue. Arrows indicate the flow of the conversation and the potential action(s) a bot or user could take.

Directions
You will be provided 2 variants of dialog flows for each conversation domain. Keeping in mind how a regular task-oriented chatbot might work, rate each flow on a scale of 1-5 on each of the four metrics: *domain (or topic) coverage*, *conclusiveness*, and *coherence*. You don't have to explain your answers.

Examples: Please refer to the document here for examples of a few flows along with some sample ratings.

Note: The annotator is recommended to come up with a rough working on what actions they believe a task-oriented dialog system for a given domain should "have", even before looking at the provided flows. This will help judge better when evaluating the provided flows for domain coverage.

General Rubric

Domain (or Topic) Coverage (1-5):

- Score 1: The flow is generic and barely covers any relevant aspects of the domain.
- Score 3: The flow covers key aspects of the domain but still misses some of them
- Score 5: The flow comprehensively covers all major and relevant aspects of the domain or topic, providing a thorough and detailed exploration.

Conclusiveness (1-5):

- Score 1: Conversations in the flow often end abruptly or leave the main query unresolved, leading to dissatisfaction.
- Score 3: Conversations tend to lead toward a resolution, but some paths may still end with questions or lack finality
- Score 5: Each conversation path leads to a clear and satisfactory conclusion or task completion, ensuring user queries are fully addressed.

Coherence (1-5):

- Score 1: The flow of conversation is disjointed or confusing, with many leaps and complex connections that disrupt understanding.
- Score 3: The conversation flow is natural for the most part but does have some non-logical paths or jumps
- Score 5: The conversation flows logically and naturally from one point to the next, with all parts making sense in the context and enhancing comprehension.

Figure 8: Evaluation Instructions for Human Annotators