Generative Induction of Dialogue Task Schemas with Streaming Refinement and Simulated Interactions

James D. Finch and **Yasasvi Josyula** and **Jinho D. Choi**
Department of Computer Science
Emory University
Atlanta, GA, USA
{jdfinch, yasasvi.josyula, jinho.choi}@emory.edu

Abstract

In task-oriented dialogue (TOD) systems, Slot Schema Induction (SSI) is essential for automatically identifying key information slots from dialogue data without manual intervention. This paper presents a novel state-of-the-art (SotA) approach that formulates SSI as a text generation task, where a language model incrementally constructs and refines a slot schema over a stream of dialogue data. To develop this approach, we present a fully automatic LLM-based TOD simulation method that creates data with high-quality state labels for novel task domains. Furthermore, we identify issues in SSI evaluation due to data leakage and poor metric alignment with human judgment. We resolve these by creating new evaluation data using our simulation method with human guidance and correction, as well as designing improved evaluation metrics. These contributions establish a foundation for future SSI research and advance the SotA in dialogue understanding and system development.

1 Introduction

Task-Oriented Dialogue (TOD) systems rely on a *slot schema* to define the key types of information used to represent the state of the dialogue as it progresses towards task completion. Traditional approaches to slot schema creation require manual curation, which is both time-consuming and difficult to scale across domains (Rastogi et al., 2020). Slot Schema Induction (SSI) has emerged as a solution, enabling automatic discovery of these slots from unlabeled dialogue data (Min et al., 2020). This task plays a crucial role in advancing TOD research by reducing the need for manual schema creation, improving dialogue state tracking (Rana et al., 2025), and facilitating automated analysis of dialogue structure (Qiu et al., 2022).

Despite the promise of SSI, current methods primarily rely on clustering dense embeddings of slot values, which may lose nuanced information about the relationship of each value to the overall dialogue (Finch et al., 2024; Yu et al., 2022). Furthermore, these methods typically require large amounts of dialogue data up-front, which is not available for most dialogue systems until after their deployment. In this paper, we present a novel approach to SSI that breaks away from clustering slot-value embeddings by casting SSI as a text generation task that incrementally constructs and refines a slot schema over a stream of dialogue data. We develop generative models for our approach using both fine-tuning and prompting techniques, where the model is tasked to create new slots that capture key values discovered in the dialogue, while also tracking the values of previously-discovered slots to maintain a consistent schema.

To develop this new method, we identify and address several critical challenges for training and evaluating SSI models. First, there is a lack of TOD data that covers a wide range of task domains, making it difficult to improve and evaluate the generalizability of SSI to novel settings. We address this by developing DOTS, a fully automatic LLM-based simulation method to create TOD data with task schemas and state labels. Second, we conduct analyses that reveal flaws in existing SSI evaluation due to benchmark data leakage and poor agreement between performance metrics and human judgment. These are addressed by constructing new evaluation data using our DOTS TOD simulation method with expert guidance and correction, and developing new evaluation metrics with superior agreement to human judgment. Using our new training data and evaluation setup, we conduct experiments comparing several variants of our method and previous approaches demonstrating that our approach achieves a new state of the art in SSI. The code, models, and data for

Transactions of the Association for Computational Linguistics, vol. 14, pp. 81–99, 2026. https://doi.org/10.1162/TACL.a.66
Action Editor: Chloe Clavel. Submission batch: 3/2025; Revision batch: 6/2025; Published 1/2026.

our approach is publicly available at https://github.com/emorynlp/UnifiedDSI.

2 Related Work

Dialogue Schema Induction TOD dialogue understanding involves creating structured representations to capture semantic information in dialogue text (Liang et al., 2024c; Budzianowski et al., 2018). Traditionally, these representations are manually defined for specific dialogue tasks, a labor-intensive process highlighting the need for automation (Agrawal et al., 2024; Liang et al., 2024c). Automating structured representation induction has been explored in various NLP tasks, including event schema induction (Dror et al., 2023; Mondal et al., 2025), text categorization (An et al., 2024; Liang et al., 2024b), intent recognition (Liang et al., 2024a; Rodriguez et al., 2024), and dialogue flow modeling (El Hattami et al., 2023; Agrawal et al., 2024; Choubey et al., 2025). Since our goal is inferring state representations for TOD, we focus on the SSI task in this work.

SSI conventionally involves two main steps: (1) extracting candidate values from dialogue text, and (2) clustering their semantic representations into slot types. Prior studies (Hudeček et al., 2021; Wu et al., 2022; Qiu et al., 2022) use NLP tagging models for extraction, embedding candidates with an encoder such as BERT (Devlin et al., 2019), then clustering them using methods like similarity thresholding (Hudeček et al., 2021) or iterative fine-tuning (Wu et al., 2022). Some works use domain-general tools such as value taggers (Qiu et al., 2022), PLM attention distributions (Yu et al., 2022), or slot-value generators (Finch et al., 2024) to discover new slot values. These methods all rely on clustering of dense value embedding vectors, which we hypothesize is suboptimal for representing slot semantics for SSI. Thus we propose a novel text generation approach for SSI that does not rely on embedding-based value clustering.

Synthetic Data Generation An ongoing challenge in TOD research is the scarcity of high-quality data that spans a diverse set of task domains. MultiWOZ (Budzianowski et al., 2018) and SGD (Rastogi et al., 2020) are currently the largest and most popular datasets by far. MultiWOZ collects human-human dialogue via a Wizard-of-Oz setup, while SGD uses rule-based dialogue simulation from handcrafted schemas with human paraphrasing. These datasets cover only 5 and 16 domains, respectively, due to their reliance on costly human annotation.

To overcome this limitation, prior work has explored data augmentation by synthesizing dialogues from human-provided task specifications, such as dialogue flows or schemas (Campagna et al., 2020; Aksu et al., 2021, 2022; Mehri et al., 2022; Mohapatra et al., 2021; Kim et al., 2021; Wan et al., 2022). More recently, LLM-based methods have been proposed to fully automate dialogue generation, eliminating human involvement in both task specification and dialogue creation (Li et al., 2023; Finch and Choi, 2024). Finch and Choi (2024) further automate state annotation in the synthesized dialogues but produce inconsistent slot schemas and noisy annotations. To the best of our knowledge, we are the first to achieve fully automated TOD simulation with a diverse set of consistent ground-truth slot schemas.

Evaluation Benchmark Leakage Concerns are mounting over the leakage of evaluation benchmarks to language models, such as OpenAI's GPT (Balloccu et al., 2024; Golchin and Surdeanu, 2024; Ranaldi et al., 2024; Xu et al., 2024). Notably for the purpose of evaluating slot schema induction approaches, Balloccu et al. (2024) revealed that the MultiWOZ 2.4 test set has been leaked to GPT. The discovery of data leakage complicates the evaluation of slot induction methods on existing datasets, as models have likely already seen the correct answers during training, making their performance unreliable as a true measure of generalization. In Section 5.1, we provide quantitative verification of MultiWOZ and SGD benchmark leakage to popular LLMs and introduce a new, unexposed evaluation dataset to overcome this challenge.

3 Approach

This work proposes using language model text generation to incrementally construct and refine a slot schema from a stream of dialogue data. Given a partial slot schema $\mathbb{S}_{t-1}$, and the last dialogue D in the data stream, our approach infers an updated schema $\mathbb{S}_t$ and dialogue state S_t for turn T_t. By initializing an empty schema $\mathbb{S}_0 = \emptyset$, this approach can be used for SSI by iterating over any dialogue datset, with the final schema $\mathbb{S}_{final}$ serving as the overall schema prediction.

Each updated schema $\mathbb{S}_t$ and state S_t are inferred jointly as a single generated text sequence

 # Key Information Types

Garden Layouts
* *style:* The preferred style of the garden layout.
* *features:* Special features included in the layout.
* *maintenance_level:* The level of maintenance required.

Plant Selections
* *type:* The type of plant, such as Flower, Shrub, Tree, or Grass.
* *color:* The color preference for the plant's blooms or foliage.

 # Dialogue
Gardener: I'm looking for a small desert garden layout with a fountain; can you help?
...
Gardener: Could you help me find some pink flowers that thrive in full sun for my garden?
Landscaper: Sure! How about the Desert Rose or Ice Plant? They both have pink flowers and thrive in full sun!
Gardener: I love the sound of the Desert Rose! Can you tell me about its water requirements?
Landscaper: The Desert Rose has medium water requirements, so it needs a bit more moisture than some other desert plants.
Gardener: I see, would it be possible to find a pink flower with low water requirements instead?

Identify Key Information Values from the Dialogue.

 # Key Information Values

Garden Layouts
* *style:* desert
* *features:* fountain
* *maintenance_level:* low

Plant Selections
* *type:* Flower
* *color:* Pink
* *sunlight:* Full Sun
- the plant's sun requirements

Figure 1: Example token sequence from DOTS (§ 4) used to train SSI model M. [YELLOW]: dialogue context D, [BLUE]: slot schema $\mathbb{S}_{t-1}$, [ORANGE]: short instruction, [GREEN]: predicted state S_t.

using a language model M. An example sequence from our training data is shown in Figure 1 to exemplify the format. As shown, our approach uses a joint formulation of zero-shot DST and new slot discovery, expanding upon prior works where these tasks were addressed separately (Gupta et al., 2022; Finch et al., 2024; Dong et al., 2024). Jointly tackling DST and new slot discovery in a single generated sequence is key: By conditioning generation on previous schema $\mathbb{S}_{t-1}$, the objective of model M is to predict the values of *existing slots* whenever possible, ensuring minimal redundancy in the slot schema across various dialogues. However, the objective of M also predicts when there are important values in D that lack existing slots. In these cases, M generates new slots with descriptive natural language labels, expanding the schema coverage to appropriately capture the underlying task structure. This process also facilitates the discovery of new task domains, since all slots are identified with a task domain name d in addition to a slot name s. Given the generated text prediction at turn t, the state S_t is simply the set of domain-slot-value triples (d, s, v) parsed from the text, and the updated schema $\mathbb{S}_t$ is calculated as:

$$\mathbb{S}_t = \mathbb{S}_{t-1} \cup \{(d, s) : (d, s, v) \in S_t\}$$

While this approach is sufficient for achieving streaming schema induction, it may produce noisy slot discoveries. Without a revision or filtering mechanism, the induced schema grows monotonically as dialogues are processed, eventually leading to degenerate inferences or implementation issues when the schema length exceeds the model's handling capacity. To address this challenge, we investigate two mechanisms for revising the schema by removing or modifying a subset of its slots: Schema Revision (§ 3.1) and Slot Confidence Estimation (§ 3.2).

3.1 Schema Revision

To revise noisy slot inferences, we train language model M to additionally revise noisy schema $\mathbb{S}_t^n$ predicted by an SSI model N into ground-truth schema $\mathbb{S}_t$. This uses a sequence format similar to Figure 1, but where the target prediction sequence represents only $\mathbb{S}_t$ without state information S_t. However, a challenge to this revision training is that any SSI model N used will likely predict an $\mathbb{S}_t^n$ with an entirely different surface form compared to $\mathbb{S}_t$, even for slots with similar meaning to the ground truth (for example, naming a slot ''max price'' versus ''budget''). Therefore, training M to predict $\mathbb{S}_t$ by directly revising $\mathbb{S}_t^n$ causes a model that rewrites the schema entirely every turn, causing a high degree of variance in the inferred schema. To improve the stability of revision prediction, we focus instead on *adding noise* to ground-truth $\mathbb{S}_t$ using $\mathbb{S}_t^n$ to obtain the schema to be revised $\mathbb{S}_t^*$, which trains M to revise the subset of $\mathbb{S}_t^*$ that is predicted to be noise. We focus on three strategies of adding noise, randomly selecting between them for each training sequence:

1. No noise is added, training M to detect when the schema needs no revision.
2. A random subset of $\mathbb{S}_t^n$ is added to $\mathbb{S}_t$, training M to revise duplicate slots.
3. A random subset of $\mathbb{S}_t^n$ is combined with a random subset of $\mathbb{S}_t$, training M to rewrite, deduplicate, or add missing slots.

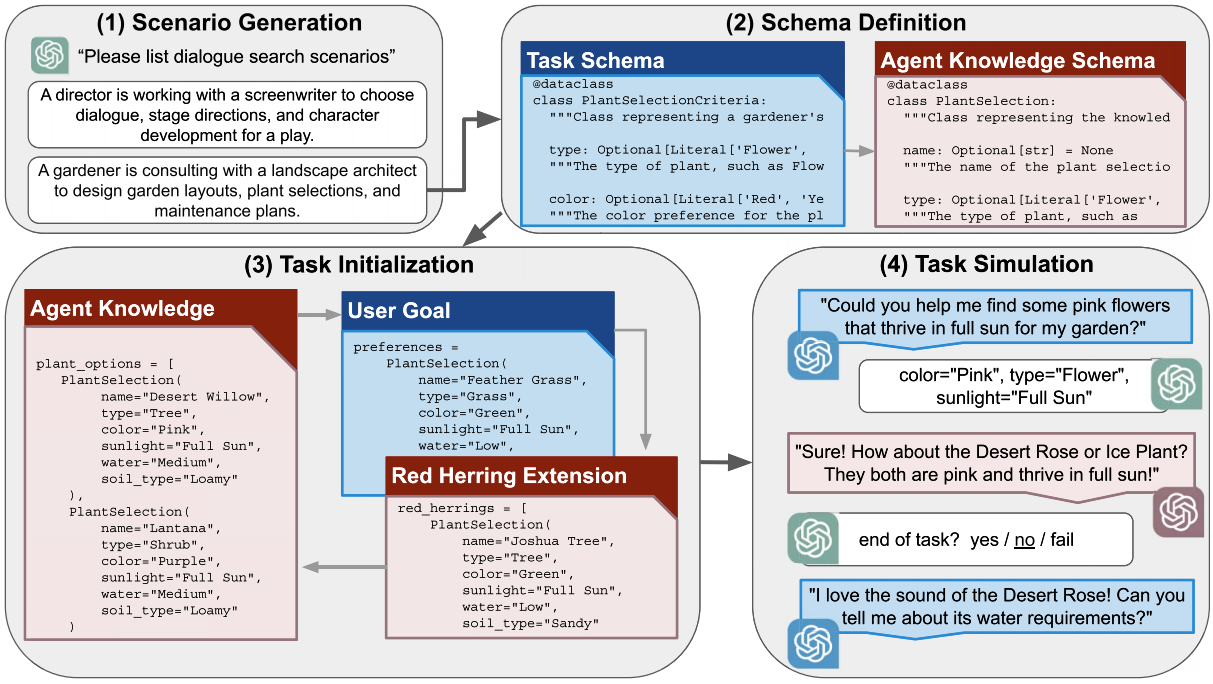

Figure 2: Overview of the dialogue simulation used to create the DOTS dataset.

3.2 Slot Confidence Estimation

In addition to modeling schema revision as another text generation subtask of our SSI approach, we observe that simply measuring the frequency with which slots in the inferred schema $\hat{\mathbb{S}}$ are assigned values in state inferences $\hat{S}$ can be used as a confidence measure of slots' suitability for representing the dialogue state. By this principle, slots that are rarely updated with values in tracked states $\hat{S}$ are likely noise and should be removed from the schema. We investigate this hypothesis by developing an alternative approach to Schema Revision.

Given a window size w, each slot in the induced schema $s \in \hat{\mathbb{S}}_t$ is removed from the schema if in recent states $\hat{S}_{t-w..t}$ it was assigned a value $c < \tau$ times, where τ represents a confidence threshold and c is the number of times its value was updated. This simple approach has the limitation of including important density-based parameters, similar to previous work's reliance on HDBSCAN clustering (Finch et al., 2024; Yu et al., 2022); however, we demonstrate in our experiments (§ 7) that it reliably filters out noisy slot discoveries.

4 Task-Oriented Dialogue Simulation

Given the limited task coverage of existing TOD training and evaluation data, we investigate a TOD simulation method inspired by Rastogi et al. (2020) and Finch and Choi (2024) to automatically generate diverse dialogues with high quality schemas and state annotations. Summarized in Figure 2, the method is broken down into four parts: Scenario Generation, Schema Definition, Task Initialization, and Task Simulation. Each of these parts is performed automatically using an LLM with zero-shot instruction. Inspired by King and Flanigan (2023), many of our prompts instruct the LLM to write a Python script with a dataclass definition or object instantiation, which we find to be a reliable method for generating structured data such as slot schemas and dialogue state labels. We use GPT-4o and GPT-4o-mini for all simulation steps.[1]

Scenario Generation Multi-domain scenarios are automatically generated as a numbered list of templated one-sentence descriptions in the form: *''<user> is getting help from <agent> in order to <A>, <B>, ...''*. Each listed sentence represents a scenario σ with speaker tags for user U and agent A roles, and an ordered list of task domains T. Generated scenarios are similar in spirit to the task settings in existing datasets like MultiWOZ and SGD. Scenarios such as travel booking with Flights and Hotel domains are generated, in

[1] gpt-4o-2024-08-06 and gpt-4o-mini-2024-07-18.

addition to many niche task settings such as garden planning (as in Figure 2) or library book checkout.

Schema Definition For each task $t \in T$ in a scenario σ, the structure of the task is defined by generating a slot schema $\mathbb{S}$ and agent knowledge schema $\mathbb{K}$. The slot schema $\mathbb{S}$ is generated first as a python dataclass based on the scenario description σ and task label t, where fields in the dataclass each represent a type of preference or requirement that user U might bring to the task. The agent knowledge schema $\mathbb{K}$ is similarly generated based on slot schema $\mathbb{S}$, t, and σ, to represent the structure of actual knowledge held by agent A. For example, if $\mathbb{S}$ has a slot called ''max price'' representing a preference, $\mathbb{K}$ should have a corresponding field like ''price'' to represent an actual value.

Task Initialization To begin simulating each TOD dialogue for a scenario σ, the schemas of its first task $t = T_1$ are used to initialize the knowledge K of agent A and the goal G of user U. K is generated as a list of Python objects constructed using the dataclass representation $\mathbb{K}$, and is meant to serve as a repository of candidate solutions to t. From this list, one item is randomly selected to serve as an ''ideal'' solution K_i. The task goal G is then generated to represent the preferences and requirements of user U based on the ideal K_i by instantiating task schema $\mathbb{S}$. A random subset of slots in G are cleared to represent no preference. To increase the difficulty of the task, a list of red herrings H is then created to extend K with items similar to G. Finally, ideal solution K_i may be removed from K a random 50% of the time to cover the realistic situation of no optimal solution. In these cases, the user U may need to change or compromise their preferences to finish the task.

Task Simulation After initializing a task t, alternating dialogue generation representing utterances for U and A builds the dialogue D. Because dialogue generation for U is conditioned only on D and G, while generation for A only sees D and K, the two simulated agents must converse to share goal and knowledge information, making a non-trivial and fairly realistic interaction. The dialogue generation prompt encourages short responses to avoid all relevant information being shared on the first turn. After each user turn, the current dialogue state is tracked given D and $\mathbb{S}$ by instructing the LLM to instantiate the schema dataclass to represent all preference information shared by U. The task simulation is detected by prompted end-of-task classification that checks D after every agent turn. When the end of the task is detected, the dialogue either continues by initializing the next task in T for scenario σ, or ends if no tasks remain.

Data	Dial.s	Turns	Dom.s	Slots	Values
MWOZ	8,420	104,916	5	31	56,668
SGD	16,142	329,964	16	115	164,982
DOT	5,015	100,471	1,003	173,572*	487,460
DOTS	2,771	88,240	787†	6,810†	44,120

Table 1: Training split statistics of MultiWOZ, SGD, DOT (Finch and Choi, 2024), and DOTS. * Slots are not semantically unique. † Slots are unique per domain, but domains sometimes overlap.

4.1 DOTS Training Data

Using the presented DOTS simulation method, we generate a TOD dataset in order to train a slot discovery model that uses the approach described in Section 3. The training data is created with $N = 300$ scenarios, 18 of which are manually removed due to domain overlap with either MultiWOZ or the 10 scenarios of the DOTS test set described in the next section (§ 5.1). Given the generated scenario descriptions and schemas for the remaining 282 scenarios, the Task Initialization and Simulation steps generate 10 dialogues per scenario. 49 dialogues are lost due to syntax issues or other errors in generated output. The final 2,771 simulated dialogues are summarized and compared to other TOD training data in Table 1.

5 Evaluation

This work addresses two critical issues for current SSI benchmark evaluation. First, we observe that TOD data has leaked into the training of the most popular base models. This issue is presented and addressed in Section 5.1 with a new test set for SSI, created using the DOTS TOD simulation pipeline with manual refinement and correction. Second, the metrics used to evaluate the previous state of the art (SotA) SSI approaches heavily overestimate performance and disagree strongly with human judgment. Section 5.2 addresses this by presenting a validation analysis of the metrics used in previous work and a new metric with vastly superior agreement to human judgment.

	MultiWOZ			SGD		
Model	**P**	**R**	**F1**	**P**	**R**	**F1**
Llama-3B	25.0	16.7	20.0	35.6	9.5	15.0
Llama-8B	45.2	46.7	45.9	71.2	21.9	33.5
Claude-3.5S	100.0	100.0	100.0	74.3	59.8	66.2
GPT-4o	100.0	100.0	100.0	84.2	50.3	63.0

Table 2: Ability of various models to recall the slot schemas of MultiWOZ and SGD from their parametric knowledge alone when instructed, providing evidence of benchmark data leakage for SSI.

5.1 New TOD Test Data

During pilot experiments, GPT-4o, Claude 3.5-Sonnet, and Llama 3.1 and 3.2 variants were observed to be capable of predicting the slot schemas of SGD and MultiWOZ *without any dialogue data present in their context*, merely an instruction to recall the slot schema given the name of the dataset. To quantify benchmark data leakage into these models' training data with respect to SSI, we conduct a human evaluation where one of the authors prompts various base models to recall the slot schema of either MultiWOZ or SGD from their parametric knowledge. Table 2 reports the Precision, Recall, and F1 scores of the slots recalled by each model compared to the gold schema, revealing that all tested models are compromised.

Given that no SSI benchmark data are available whose schemas have not been leaked into the training of relevant base models, a new evaluation dataset is constructed to fill this gap. Ideally, the test data would be collected using recordings of a wide range of real-world naturalistic dialogue scenarios with expert-annotated task schemas and dialogue states. However, such data collection is extraordinarily expensive and difficult to achieve. Thus, the methods used to create previous benchmark data MultiWOZ and SGD were semi-automatic. The creation of our new data broadly follows SGD's approach, but takes advantage of LLM's high-quality dialogue generation ability using our DOTS simulation pipeline:

Test Domains are created using 10 handcrafted scenario descriptions written by one of the authors. 5 scenarios include 2 domains and 5 include 3, making 25 unique task domains in total. Using a handcrafted approach, the scenarios are guaranteed to reflect plausible TOD applications that are also unique among each other and compared to domains in MultiWOZ and SGD.

Domain Schemas are created by feeding handcrafted scenario descriptions to our Schema Definition pipeline. To ensure the resulting schemas are a valid reflection of the task scenario, all 25 domain task schemas are manually corrected by 2 of the authors. One of the authors, an undergraduate with experience in NLP research, performed the majority of the corrections, which were then reviewed for correctness by the other author who is a postdoctoral researcher. In general, the schemas generated by our pipeline are high-quality: from 212 original slots, 4 new slot types were created to address potential constraints for successful task completion that were missing, 8 nonsensical slot types were removed, and 2 slots were revised to provide additional clarity in their name and description, resulting in a final total of 208 slots across the 25 domains. Appendix A shows our correction guideline.

State-Annotated Dialogues are created by feeding the corrected schemas into the Task Initialization and Task Simulation stages of the simulation pipeline. One hundred dialogues were simulated per scenario. From these 1,000 dialogues, 30 per scenario were then manually cherry-picked and then corrected. Hand-selecting a subset of 300 dialogues from a larger set of 1,000 allows avoiding degenerate dialogues, since in rare instances, simulated dialogues stall on a task or end prematurely if the end-of-task classification step fails. This selection process was fully manual, relying on the judgment of the annotators to filter out degenerate or unnatural dialogues, and ensure selected dialogues have good coverage of their respective task schemas.

After 300 dialogues were cherry-picked to include in the test set, one of the authors proceeded to correct the dialogues by fixing any incorrect or hallucinated slot values and improving awkward dialogue responses where needed. After dialogues were corrected, another author reviewed the corrections and made a small number of minor revisions. The entire correction process for 300 dialogues took approximately 10–15 hours, averaging 2–3 minutes per dialogue. The guideline used for the corrections can be viewed in Appendix B. Among the final set of 300 dialogues, 7.2% of turns were revised to improve

Data	Dial.s	Turns	Dom.s	Slots	Values
MWOZ	999	13,737	5	30	7,368
SGD	4,201	84,594	18	106	42,297
DOTS	300	7,844	25	208	3,922

Table 3: Statistics of SSI test data.

naturalness, and 4.3% of slot-value pairs were corrected due to errors produced by the automatic state annotator.

Although the DOTS test set is smaller at a little more than half the size of MultiWOZ, it has the highest degree of domain diversity among the three evaluation data compared in Table 3 with almost twice as many slot types as SGD. This diversity makes it more suitable as a test set for SSI, since approaches are tested on their ability to induce schemas across a wide range of diverse dialogue tasks. In fact, because DOTS includes 10 unique and disjoint multi-domain application scenarios, it facilitates estimating the *expected performance* for schema induction on any novel scenario by averaging performance across independent per-scenario evaluations. This allows more realistic and reliable estimation of SSI performance than was possible with previous evaluation data MultiWOZ and SGD, which arguably each only represent a single application scenario. Our experiments thus follow this paradigm of estimating expected performance by averaging per-scenario evaluation metrics.

5.2 Improved Evaluation Metrics

Evaluation of SSI aims to measure the quality of an induced set of slots P by matching it against a gold reference slot schema G (Finch et al., 2024; Yu et al., 2022). Each predicted slot $p \in P$ and gold slot $g \in G$ are represented as sets of context-value pairs (c, v), where c is a particular dialogue context and v is a value filling the slot. Given predicted slots and gold slots P and G, an SSI evaluator defines a mapping $M : P \rightarrow G$ that associates each predicted slot $p \in P$ to the reference slot $g \in G$ that has the highest similarity score $S(p, g)$:

$$M^* = \{(p, g) \mid p \in P, g = \arg\max_{g \in G} S(p, g)\}$$

Previous SotA SSI approaches are evaluated automatically (Finch et al., 2024; Yu et al., 2022), where S is implemented as S_{BERT} by computing the centroid of each induced and gold reference slot cluster using BERT encodings (Devlin et al., 2019) of their values. Each induced cluster p is mapped to the gold slot g whose cluster centroid is nearest by cosine similarity, or not mapped if there is no match of 80% similarity or higher. See Finch et al. (2024) for further detail.

We find this automatic matching method based on embedding similarity is extremely noisy and does not reflect human judgments of the slot mapping. Thus, we propose the alternative similarity function $S_{exact}(p, g)$ calculated as the precision of values in p that exactly match (caseless) those in g:

$$S_{\text{exact}}(p, g) = \frac{|p \cap g|}{|p|}$$

Predicted slots are matched to the most similar gold slot as long as a threshold of 50% precision is met, $S_{exact}(p, g) \geq 0.5$, with predicted slots remaining unmatched if there is no gold slot meeting this criterion.

Validation Study To evaluate the performance of each automatic matcher, we conduct a validation experiment using human-annotated slot mappings, denoted as M_h. We consider four experimental settings derived from evaluating both the prior SotA SSI approach (Finch et al., 2024) and the best-performing model trained in this work, using the test splits of DOTS and MultiWOZ 2.4 (Ye et al., 2022). Specifically, a Llama-8B model trained on the DOTS train split with Slot Confidence estimation (§3.2) represents our presented approach based on its performance shown in Section 7.

For each setting, a human expert (one of the authors) manually constructed M_h based on the predicted slots P. These human-created mappings serve as ground-truth references for evaluating automatically generated mappings produced using the BERT-based similarity function S_{BERT} (from prior work) and the proposed similarity function S_{exact}. The guideline for constructing the mapping is presented in Appendix C.

To assess the reliability of the human annotations, a second author independently constructed M_h for the same settings. The resulting inter-annotator agreement yielded an average Cohen's kappa of 0.62 and a raw agreement rate of 72%, indicating moderate to substantial agreement. Disagreements primarily occurred when a predicted slot's example values have ambiguous

Data	Approach	S_{BERT}	S_{exact}
MWOZ	Finch et al. (2024)	18.8	**69.4**
MWOZ	Ours	54.5	**64.4**
DOTS	Finch et al. (2024)	35.3	**71.8**
DOTS	Ours	70.5	**71.4**

Table 4: Accuracy of predicted-to-gold slot mappings produced using S_{BERT} and S_{exact} metrics for the previous SotA SSI method and the best-performing approach in this work: Llama-8B model trained on DOTS data using the final state representation and slot confidence approach.

membership in a corresponding gold slot's definition. Such ambiguity in mapping predicted to gold slots is often the result of small differences in definition scope or data type between the gold and predicted slot.

Table 4 presents the percentage of predicted slots that were correctly mapped using S_{BERT}, compared to the percentage of correctly mapped slots using S_{exact} for each of the 4 settings. It is clear that S_{exact} far outperforms S_{BERT} in mapping predicted slots to gold slots. Although S_{exact} only agrees with human judgement about $+1\%$ to $+10\%$ more than S_{BERT} when evaluating our strongest trained model, the advantage of S_{exact} is pronounced when evaluating weaker SSI models such as the previous SotA, showing around a $+35\%$ to $+50\%$ improvement. Disagreements between M_h and $\hat{M}$ predicted using S_{exact} were, as expected, usually due to overly strict similarity estimation producing no match in cases where the human found a match, or due to impure slot clusters in P combining the semantics of multiple slots in G, which creates ambiguous mapping decisions.

Metrics Table 5 presents the formulas for the evaluation metrics used in our experiments, which are calculated given mapping $\hat{M}$ constructed by our improved automatic matcher. Note that, unlike the presented metric equations, previous work allows multiple predicted slots to be mapped to the same gold slot (Finch et al., 2024; Yu et al., 2022). This results in vastly inflated scores when calculating precision, and encourages approaches that focus on optimizing recall by predicting an order of magnitude more slots than what appear in gold schemas. Our metrics adjust the precision calculation to count each redundant slot prediction as a precision error.

	Precision	Recall
Slot	$\frac{\lvert\{g:(p,g)\in M\}\rvert}{\lvert P\rvert}$	$\frac{\lvert\{g:(p,g)\in M\}\rvert}{\lvert G\rvert}$
Value	$\frac{\sum_{(p,g)\in M}\lvert p\cap g\rvert}{\sum_{(p,g)\in M}\lvert p\rvert}$	$\frac{\sum_{(p,g)\in M}\lvert p\cap g\rvert}{\sum_{(p,g)\in M}\lvert g\rvert}$

Table 5: Precision and Recall formulas for evaluation metrics of induced slots and discovered values.

6 Experiments

Experiments use the DOTS evaluation data and refined metrics from Section 5. Since slot induction aims to induce a schema for tracking dialogue state in novel settings, we assess performance across multi-domain scenarios by evaluating slot induction on each of the 10 test scenarios independently and reporting the average for each metric. All experiment results are also averaged over 3 replicates to address any stochastic variance like dialogue stream order. Each evaluation in our experiments is a unique combination of approach (§ 6.1), state representation (§ 6.2), training data (§ 6.3), base model (§ 6.4), and evaluation data (§ 6.5). To avoid combinatorial explosion of settings, some evaluation setting combinations are excluded based on one or more of their features yielding poor performance compared to alternatives.

6.1 Approaches

REVISION is our main streaming slot induction approach using the revision mechanism (§ 3.1). Revision training data is obtained from the outputs of a SLOT CONF model trained on a small development version of DOTS separate from the training and evaluation splits, thus producing noisy outputs that satisfy the revision setup.

SLOT CONF is our main streaming slot induction approach using slot update counts to estimate confidence in each slot for filtering noisy discoveries (§ 3.2). The window length is set to 10 dialogues and the confidence threshold to 1 update.

FIFO is a baseline simplifying SLOT CONF, which filters out the least-recently-filled slots when the predicted schema size exceeds 100 slots.

PRIORITY is another baseline of SLOT CONF that maintains global slot update counts, removing the least frequently updated slots when the schema reaches 100 slots.

EMBED follows prior slot induction work from Finch et al. (2024) where SBERT embeddings of induced slots are clustered to predict slot schemas. EMBED models trained on DOT embed slot name + value, while models trained on DOTS perform better when embedding predicted slot descriptions. Clustering uses HDBSCAN (McInnes et al., 2017) with hyperparameters selected from a grid search optimizing silhouette score (Yu et al., 2022).

Note that, because our streaming slot induction approach (§ 3) incrementally updates schema predictions, all experiments with streaming methods use a two-pass setup. In pass 1, the final schema is determined, and then pass 2 runs in ''DST mode,'' ignoring new slot discoveries, to allow the model to represent earlier states in the dialogue data with the final slot schema. This ensures a fairer comparison to clustering methods without penalizing streaming approaches for early schema revisions.

6.2 State Representations

The main approach presented in Section 3 is described to use a sequence format that predicts dialogue state representations for the most recent dialogue turn, which is similar to sequence-to-sequence dialogue state trackers (King and Flanigan, 2023; Gupta et al., 2022; Finch and Choi, 2024). However, the sequence format can be redesigned to predict only the *updates* to the dialogue state made by the most recent dialogue turn, similar to previous slot induction work (Finch et al., 2024; Yu et al., 2022), or to make slot induction predictions on the last turn of the dialogue only.

STATE predicts full dialogue states including values from the entire dialogue history, similar to DST work, for all turns in each dialogue.

UPDATE predicts only dialogue state updates from the most recent turn, following prior slot induction work. Full dialogue state predictions for each turn are deterministically recovered after generating predictions for all turns of any given dialogue by initializing a state with no filled slots and iteratively applying each turn's predicted update.

FINAL predicts only the final state of each dialogue during schema induction, which may reduce noisy new slot discoveries caused by a lack of context for how the dialogue progresses. Once the schema is finalized at the end of pass 1, dialogue states are predicted for each turn in pass 2, identical to the STATE method.

6.3 Training Data

DOTS is the new data presented in Section 4.1.

DOT (Finch and Choi, 2024) lacks ground-truth schemas, preventing training of streaming approaches, but is used to train variants of EMBED.

SGDX augments SGD (Rastogi et al., 2020), the most domain-diverse existing dataset with ground-truth schemas, by replacing each training sequences' slots and descriptions with one of their six SGD-X (Lee et al., 2022) variants.

6.4 Base Models

LLAMA-8B/3B/1B Instruct 3.1/3.2 variants (Dubey et al., 2024) are fine-tuned using QLoRA with rank 1 and alpha 2, applying adapters to all linear layers, following findings that rank has little impact when applied consistently across all linear layers (Dettmers et al., 2023). Training uses the Adam optimizer with a learning rate of 10^{-4}, batch size of 8, and 30,000 steps.

T5-3B (Raffel et al., 2020) is evaluated for direct comparison with previous work, using the publicly released model from Finch et al. (2024) that was trained on DOT.

CLAUDE 3.5-Sonnet (Anthropic, 2024) is assessed. As a closed-access model, it is evaluated via API with zero-shot instruction for slot induction instead of fine-tuning.

The GPT-* family of models are excluded from our experiments because they were used to generate the evaluation data, which is likely to bias their SSI performance and results (Panickssery et al., 2024).

6.5 Test Data

DOTS is the test split of the new, human-corrected data presented in Section 5.1.

MultiWOZ 2.4 test split (Ye et al., 2022) is also included for evaluation for further analysis and comparison to previous work. Experiments follow Finch et al. (2024) and Yu et al. (2022)

				Slot			Value		
Data	**Model**	**Approach**	**State**	**P**	**R**	**F1**	**P**	**R**	**F1**
DOT	T5	Embed	U	22.9	39.5	21.1	63.3	20.3	27.6
DOT	Ll-8B	Embed	U	45.3	20.3	22.0	73.4	45.5	53.6
DOTS	Ll-8B	Embed	U	39.9	37.6	25.5	73.3	29.9	38.0
SGD-X	Ll-8B	Revision	F	51.9	19.1	27.5	71.4	55.5	61.5
DOTS	Ll-1B	Slot Conf	F	32.4	49.2	38.6	71.6	49.4	57.2
SGD-X	Ll-8B	Slot Conf	F	47.5	43.8	44.7	75.1	50.5	59.6
DOTS	Ll-8B	Embed	F	**74.3**	39.4	46.2	76.3	77.4	74.8
DOTS	Ll-8B	Revision	S	43.3	**73.5**	53.2	80.4	50.9	61.4
DOTS	Ll-8B	Slot Conf	S	45.9	70.5	55.0	82.7	54.7	66.5
DOTS	Ll-8B	Revision	F	50.4	70.5	57.8	80.5	60.0	68.2
DOTS	Ll-8B	Revision	U	52.3	68.3	58.2	**85.6**	44.3	56.9
DOTS	Ll-3B	Slot Conf	F	54.5	64.8	58.5	77.9	64.1	69.7
DOTS	Ll-8B	Slot Conf	U	50.9	73.1	59.4	84.1	45.4	57.8
	Claude	Revision	F	64.3	56.5	59.7	81.1	**93.5**	**86.5**
DOTS	Ll-8B	Priority	F	54.0	70.8	60.6	81.4	63.8	71.2
DOTS	Ll-8B	Fifo	F	55.2	72.1	62.0	81.5	64.7	71.7
DOTS	Ll-8B	Slot Conf	F	62.5	72.6	66.8	82.5	69.2	74.9
	Claude	Slot Conf	F	74.2	65.7	**69.2**	80.8	88.9	84.3

Table 6: Average Precision/Recall/F1 (**P/R/F1**) on induced slots and discovered values of the DOTS test set, sorted by Slot F1. S: STATE, U: UPDATE, F: FINAL.

by evaluating each SSI approach's ability to induce all informable slots in the gold schema for fair comparison to Finch et al.'s (2024) model. Note that, as demonstrated in Section 5.1, this MultiWOZ 2.4 test data has been leaked into the pretraining of many popular base models, including the ones we use in our experiments. Therefore, performance on this test set may not generalize to real-world SSI.

7 Results

Comparison of Approaches Table 6 presents experiment results on DOTS. Of the three main approach categories, the SLOT CONF based streaming methods appear to perform the best, with SLOT CONF predicting FINAL states achieving both the highest Slot F1 (66.8) and Value F1 (74.9). The REVISION streaming methods performed second-best, although they are outperformed by even the baseline SLOT CONF approaches: FIFO and PRIORITY. Furthermore, the EMBED approach from previous work performed worst, which supports our hypothesis that using embedding clustering for slot induction is suboptimal, although the FINAL state variant achieved high Slot Precision (74.3) with high-quality clusters at a Value F1 of 74.8. In general, prediction of FINAL states appears more reliable than STATE or UPDATE alternatives, across all methodologies.

Comparison of Domain-Diverse Data The utility of DOTS as a training resource is assessed by comparing models trained with DOTS against those trained with DOT, which is a similarly domain-diverse dataset but created using different data generation methods. The results show that the main limitation of the DOT data as a training resource is its lack of ground-truth schemas, which are necessary to train the higher-performing SLOT CONF and REVISION approaches. As only the EMBED approach can be used with DOT training data, the highest Slot F1 reached is 22.0 by the LLAMA-8B model, indicating poor performance. Replacing DOT with the new DOTS data in the same LLAMA-8B EMBED UPDATES approach setup improves performance slightly to 25.5 Slot F1, which suggests DOTS is a superior training resource despite its smaller size. Furthermore, training an EMBED approach to predict FINAL state representations on DOTS boosts performance to 46.2 Slot F1, but this training formulation is not supported by DOT due to inconsistent slot names between different turns within the same dialogue. These results provide

				Slot			Value		
Data	**Model**	**Approach**	**State**	**P**	**R**	**F1**	**P**	**R**	**F1**
DOT	T5	Embed	U	0.9	**100.0**	1.8	90.1	1.3	2.5
DOT	T5	Embed*	U	12.5	66.7	21.1	80.3	11.5	20.2
DOTS	Ll-8B	Embed	F	36.7	34.1	30.6	59.2	60.1	57.0
DOTS	Ll-8B	Priority	F	20.0	66.7	30.8	78.0	40.0	52.9
DOT	Ll-8B	Embed	U	21.8	63.3	32.5	69.4	14.0	23.3
DOTS	Ll-8B	Revision	S	27.7	69.7	37.4	77.0	50.9	59.4
DOTS	Ll-8B	Revision	U	28.2	66.7	39.6	79.2	32.5	46.1
DOTS	Ll-8B	Slot Conf	U	32.9	71.7	42.3	86.5	46.2	57.0
DOTS	Ll-8B	Slot Conf	S	30.6	77.5	42.5	84.3	50.5	61.5
DOTS	Ll-8B	Revision	F	32.3	70.0	44.2	74.8	70.0	72.3
DOTS	Ll-8B	Slot Conf	F	41.0	71.5	50.6	83.5	67.9	73.9
SGD-X†	Ll-8B	Slot Conf	F	58.5	68.8	60.2	84.6	62.8	70.7
	Claude	Slot Conf	F	**85.0**	58.6	69.4	**94.2**	**94.2**	**94.2**
SGD-X	Ll-8B	Slot Conf	F	70.0	82.2	**74.7**	84.2	78.3	80.3

Table 7: Average Precision/Recall/F1 (**P/R/F1**) on induced slots and discovered values of the MultiWOZ 2.4 test set, sorted by Slot F1. *Uses the HDBSCAN hyperparameters presented in Finch et al. (2024) instead of automatically optimizing silhouette score. †Domains that have significant semantic overlap with MultiWOZ (Hotels, RideSharing, RentalCars, Restaurants, and Travel) are filtered out.

evidence that the structured and simulated DOTS generation method reduces noise and improves SSI training above the method used to generate the DOT data.

Comparison of Schema-Consistent Data Comparing the efficacy of DOTS training data against the existing schema-consistent dataset SGD, we find DOTS to be the superior training resource for the novel streaming slot induction method. For both SLOT CONF and REVISION approaches, DOTS trained models achieve better performance. Fine-tuning the LLAMA-8B base model with FINAL state inferences, the SLOT CONF approach achieved 66.8 Slot F1 when trained on DOTS versus only 44.7 Slot F1 when trained on SGD-X, and the REVISION approach reached 57.8 Slot F1 trained on DOTS versus only 27.5 Slot F1 trained on SGD-X. This massive performance gap demonstrates the necessity of domain-diverse training data, echoing similar findings of Finch et al. (2024) which showed the efficacy of DOT above SGD-X for SSI due to increased domain diversity.

Fine-tuning Versus Prompting Among all methods evaluated, CLAUDE in SLOT CONF mode achieves the highest overall performance, with an average Slot F1 of 69.2 and an average Value F1 of 84.3. Our fine-tuned LLAMA-8B model performs competitively, coming within 2.5 points of CLAUDE in Slot F1 and 11 points in Value F1. Moreover, CLAUDE is not superior in all aspects, as our best fine-tuned LLAMA-8B SLOT CONF model exhibits superior Slot Recall and Value Precision. In particular, Slot Recall is often more important for SSI in practice than Slot Precision, as it is much easier to manually discard some noisy induced slots than to manually review dialogue data to recover missing inductions. These results suggest a trade-off between prompting and fine-tuning SSI models. Significantly smaller base models like Llama-8B can be fine-tuned to achieve performance similar to what larger LLMs can achieve with prompting, which makes for cost-efficient SSI inference with higher Slot Recall. On the other hand, prompting a larger LLM for SSI may afford additional control over SSI inferences and is shown to produce higher-quality slot clusters at a higher Slot Precision.

MultiWOZ Evaluation The MultiWOZ evaluation results in Table 7 confirm the overall trends observed on DOTS evaluation, with SLOT CONF streaming approaches achieving the best performance among Llama-8B variants, REVISION methods performing moderately, and EMBED being least effective. Models trained on the SGD-X

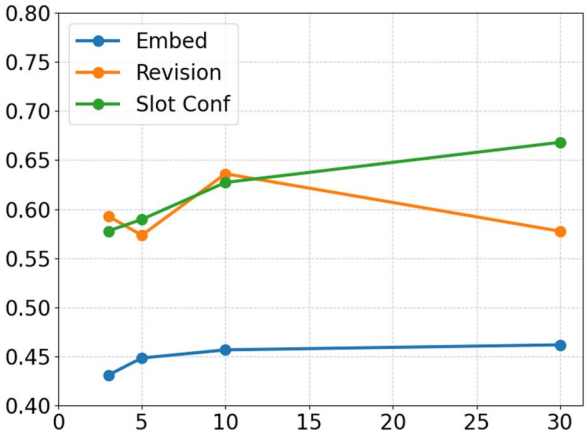

Figure 3: Slot F1 Score vs. Number of Dialogues when evaluating LLAMA-8B fine-tuned on DOTS with either the EMBED FINAL, REVISION FINAL, or SLOT CONF FINAL approaches.

dataset attain the highest MultiWOZ scores (up to Slot F1: 74.7), attributable to the structural and domain similarity between SGD-X and MultiWOZ. When domains overlapping with MultiWOZ are filtered out of the SGD-X training data, Slot F1 drops more than 14 points. The Llama-8B SLOT CONF model trained on DOTS performs well despite no training domain overlap with MultiWOZ, achieving higher Slot Recall and Value F1 than the model trained on filtered SGD-X. Models trained on DOT and DOTS have lower Slot Precision than those trained on SGD-X. This is partially due to these training data not carrying any distinction between informable and requestable slots, leading to induction of some requestable slots like ''Attraction phone number'' that count as precision errors. The distribution shift from DOTS training data to MultiWOZ testing data may also contribute to the loss of precision, although Slot Recall and Value F1 performance was similar to that achieved on DOTS evaluation data.

Low Resource Slot Discovery Figure 3 presents an analysis of SSI performance in low-resource settings where a slot schema must be induced from only 3-30 dialogues. This analysis evaluated the performance of LLAMA-8B EMBED, REVISION, and SLOT CONF, trained on DOTS with FINAL state predictions. The test set of each evaluation is a downsampled subset of DOTS test dialogues. The results suggest that SLOT CONF schema induction benefits the most from increased data, while EMBED remains relatively stagnant, and REVISION exhibits inconsistent performance trends.

	Slot		
Model	**P**	**R**	**F1**
LL-8B	76.8	**88.2**	81.8
CLAUDE	**91.3**	80.0	**84.6**

Table 8: Average Precision/Recall/F1 (**P/R/F1**) on induced slots based on human mapping of predicted-to-gold slots for best-performing models: Llama-8B SLOT CONF (**Ll-8B**) and Claude SLOT CONF (**Claude**).

Human Evaluation Results Since automatic SSI evaluation metrics do not perfectly align with human judgements we perform a human evaluation of the predicted slot schemas from our best LLAMA-8B model and CLAUDE using the method described in Section 5.2. The results are consistent with the automatic evaluation results, but both evaluated approaches benefit from less strict matching between predicted and gold slots, achieving 81.8 and 84.6 Slot F1 respectively (Table 8).

8 Discussion

Induction as Text Generation is found to be a powerful approach for slot induction. Our experiments highlight the advantages of jointly inferring newly discovered slots while tracking existing ones through streaming generation, significantly outperforming embedding-based clustering. Furthermore, integrating DST into this approach provides a simple yet effective confidence measure for each discovered slot, enabling the filtering of noisy slots as more dialogue data becomes available. Surprisingly, the text generation–based schema refinement approach, REVISION, does not improve upon the other studied refinement method, SLOT CONF, or its baselines, FIFO and PRIORITY. Manual analysis suggests the limitation of REVISION to be that it produces high variance schema inferences. This may be caused by overly local revision decisions made based on single dialogue examples, whereas the simpler SLOT CONF method benefits from confidence estimations across many dialogues. One possible direction for future work is to explore LLM-based refinement strategies that incorporate global statistical patterns, perhaps by encoding such information as part of the model's input, in efforts to further

improve and explore the capabilities of this joint approach to SSI.

TOD Data Creation remains an ongoing research challenge, but our work demonstrates the power of LLM-based simulation for constructing TOD resources for training and evaluation. The DOTS automatic data generation method is the first to create TOD data for novel diverse tasks with ground-truth task schemas. The strategy of generating dialogues as simulations from generated task schemas turns out to produce higher quality state-labeled dialogue data than the noisier automatic-annotation strategy from previous work (Finch and Choi, 2024), and allows for multi-domain and schema-consistent dialogue creation at low cost. TOD simulation methods may continue to alleviate the difficult and costly nature of collecting data, and future work may help to further improve the realism and applicability of our presented methods.

Slot Induction Evaluation benchmarks from previous work are invalidated due to the data leakage and evaluation metric issues demonstrated in our analyses (§ 5). We address these issues by providing new validated evaluation metrics and evaluation data created with human guidance and correction. However, there may be room for further improvement, since the automatically-generated nature of the DOTS evaluation data inevitably carries biases compared to the data distribution of real-world TOD settings. Although costly, investment in creating more realistic and diverse TOD datasets may be a key step forward for all TOD research, especially since other subtasks such as DST may be affected by data leakage as well.

9 Conclusion

The presented work marks a large step forward for SSI, advancing TOD research. Our findings highlight the importance of text generation-based approaches for slot induction, the value of simulation-based data creation, and the necessity of rigorous evaluation methodologies for advancing TOD research. We publicly release the new SotA slot induction models and DOTS data to facilitate future work. These contributions improve automatic TOD understanding and support dialogue system development, helping to create more natural and effective human-computer interactions.

Acknowledgments

We gratefully acknowledge the support of Hyundai Motor. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Hyundai Motor. We also thank the reviewers and action editor for their valuable feedback.

References

Stuti Agrawal, Pranav Pillai, Nishi Uppuluri, Revanth Gangi Reddy, Sha Li, Gokhan Tur, Dilek Hakkani-Tur, and Heng Ji. 2024. Dialog flow induction for constrainable LLM-based chatbots. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 66–77, Kyoto, Japan. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.sigdial-1.6

Ibrahim Aksu, Zhengyuan Liu, Min-Yen Kan, and Nancy Chen. 2022. N-shot learning for augmenting task-oriented dialogue state tracking. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1659–1671, Dublin, Ireland. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.findings-acl.131

Ibrahim Taha Aksu, Zhengyuan Liu, Min-Yen Kan, and Nancy Chen. 2021. Velocidapter: Task-oriented dialogue comprehension modeling pairing synthetic text generation with domain adaptation. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 133–143, Singapore and Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.sigdial-1.14

Wenbin An, Wenkai Shi, Feng Tian, Haonan Lin, QianYing Wang, Yaqiang Wu, Mingxiang Cai, Luyan Wang, Yan Chen, Haiping Zhu, and Ping Chen. 2024. Generalized category discovery with large language models in the loop. In *Findings of the Association for Computational*

Linguistics: ACL 2024, pages 8653–8665, Bangkok, Thailand. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.findings-acl.512

AI Anthropic. 2024. Claude 3.5 sonnet model card addendum. *Claude-3.5 Model Card*, 3(6).

Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondrej Dusek. 2024. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source LLMs. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 67–93, St. Julian's, Malta. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.eacl-long.5

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 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, Brussels, Belgium. Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-1547

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 the 58th Annual Meeting of the Association for Computational Linguistics*, pages 122–132, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.12

Prafulla Kumar Choubey, Xiangyu Peng, Shilpa Bhagavath, Caiming Xiong, Shiva Kumar Pentyala, and Chien-Sheng Wu. 2025. Turning conversations into workflows: A framework to extract and evaluate dialog workflows for service ai agents. *arXiv preprint arXiv:2502.17321*. https://doi.org/10.18653/v1/2025.findings-acl.203

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. *Advances in Neural Information Processing Systems*, 36:10088–10115.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiaoyu Dong, Yujie Feng, Zexin Lu, Guangyuan Shi, and Xiao-Ming Wu. 2024. Zero-shot cross-domain dialogue state tracking via context-aware auto-prompting and instruction-following contrastive decoding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8527–8540, Miami, Florida, USA. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.emnlp-main.485

Rotem Dror, Haoyu Wang, and Dan Roth. 2023. Zero-shot on-the-fly event schema induction. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 705–725, Dubrovnik, Croatia. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-eacl.53

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai,

Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Amine El Hattami, Issam H. Laradji, Stefania Raimondo, David Vázquez, Pau Rodríguez, and Christopher Pal. 2023. Workflow discovery from dialogues in the low data regime. *Transactions on Machine Learning Research. arXiv preprint arXiv.2205.11690*.

James D. Finch and Jinho D. Choi. 2024. Diverse and effective synthetic data generation for adaptable zero-shot dialogue state tracking. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12527–12544, Miami, Florida, USA. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.findings-emnlp.731

James D. Finch, Boxin Zhao, and Jinho D. Choi. 2024. Transforming slot schema induction with generative dialogue state inference. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 317–324, Kyoto, Japan. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.sigdial-1.27

Shahriar Golchin and Mihai Surdeanu. 2024. Time travel in llms: Tracing data contamination in large language models. In *The Twelfth International Conference on Learning Representations*.

Raghav Gupta, Harrison Lee, Jeffrey Zhao, Yuan Cao, Abhinav Rastogi, and Yonghui Wu. 2022. Show, don't tell: Demonstrations outperform descriptions for schema-guided task-oriented dialogue. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4541–4549, Seattle, United States. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.naacl-main.336

Vojtěch Hudeček, Ondřej Dušek, and Zhou Yu. 2021. Discovering dialogue slots with weak supervision. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2430–2442, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.189

Sungdong Kim, Minsuk Chang, and Sang-Woo Lee. 2021. NeuralWOZ: Learning to collect task-oriented dialogue via model-based simulation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3704–3717, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.287

Brendan King and Jeffrey Flanigan. 2023. Diverse retrieval-augmented in-context learning for dialogue state tracking. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5570–5585, Toronto, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-acl.344

Harrison Lee, Raghav Gupta, Abhinav Rastogi, Yuan Cao, Bin Zhang, and Yonghui Wu. 2022. SGD-X: A benchmark for robust generalization in schema-guided dialogue systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10938–10946. https://doi.org/10.1609/aaai.v36i10.21341

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for ''mind'' exploration of large language model society. *Advances in Neural Information Processing Systems*, 36:51991–52008.

Jinggui Liang, Lizi Liao, Hao Fei, and Jing Jiang. 2024a. Synergizing large language models and pre-trained smaller models for conversational intent discovery. In *Findings*

of the Association for Computational Linguistics: ACL 2024, pages 14133–14147, Bangkok, Thailand. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.findings-acl.840

Jinggui Liang, Lizi Liao, Hao Fei, Bobo Li, and Jing Jiang. 2024b. Actively learn from LLMs with uncertainty propagation for generalized category discovery. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7845–7858, Mexico City, Mexico. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.naacl-long.434

Jinggui Liang, Yuxia Wu, Yuan Fang, Hao Fei, and Lizi Liao. 2024c. A survey of ontology expansion for conversational understanding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18111–18127. https://doi.org/10.18653/v1/2024.emnlp-main.1006

Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11):205. https://doi.org/10.21105/joss.00205

Shikib Mehri, Yasemin Altun, and Maxine Eskenazi. 2022. LAD: Language models as data for zero-shot dialog. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 595–604, Edinburgh, UK. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.sigdial-1.55

Qingkai Min, Libo Qin, Zhiyang Teng, Xiao Liu, and Yue Zhang. 2020. Dialogue state induction using neural latent variable models. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20, pages 3845–3852. Yokohama, Yokohama, Japan. https://doi.org/10.24963/ijcai.2020/532

Biswesh Mohapatra, Gaurav Pandey, Danish Contractor, and Sachindra Joshi. 2021. Simulated chats for building dialog systems: Learning to generate conversations from instructions. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1190–1203. Punta Cana, Dominican Republic. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.findings-emnlp.103

Ishani Mondal, Michelle Yuan, Anandhavelu N., Aparna Garimella, Francis Ferraro, Andrew Blair-Stanek, Benjamin Van Durme, and Jordan Boyd-Graber. 2025. ADAPTIVE IE: Investigating the complementarity of human-AI collaboration to adaptively extract information on-the-fly. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5870–5889, Abu Dhabi, UAE. Association for Computational Linguistics.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. 2024. LLM evaluators recognize and favor their own generations. *Advances in Neural Information Processing Systems*, 37:68772–68802. https://doi.org/10.52202/079017-2197

Liang Qiu, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. Structure extraction in task-oriented dialogues with slot clustering. ArXiv:2203.00073 [cs]. https://doi.org/10.48550/arXiv.2203.00073

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Mansi Rana, Kadri Hacioglu, Sindhuja Gopalan, and Maragathamani Boothalingam. 2025. Zero-shot slot filling in the age of LLMs for dialogue systems. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 697–706, Abu Dhabi, UAE. Association for Computational Linguistics.

Federico Ranaldi, Elena Sofia Ruzzetti, Dario Onorati, Leonardo Ranaldi, Cristina Giannone, Andrea Favalli, Raniero Romagnoli, and Fabio Massimo Zanzotto. 2024. Investigating the impact of data contamination of large language models in text-to-SQL translation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13909–13920,

Bangkok, Thailand. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.findings-acl.827

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05): 8689–8696, https://doi.org/10.1609/aaai.v34i05.6394

Juan A. Rodriguez, Nicholas Botzer, David Vazquez, Christopher Pal, Marco Pedersoli, and Issam H. Laradji. 2024. Intentgpt: Few-shot intent discovery with large language models. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

Dazhen Wan, Zheng Zhang, Qi Zhu, Lizi Liao, and Minlie Huang. 2022. A unified dialogue user simulator for few-shot data augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3788–3799, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.findings-emnlp.277

Yuxia Wu, Lizi Liao, Xueming Qian, and Tat-Seng Chua. 2022. Semi-supervised new slot discovery with incremental clustering. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6207–6218, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.findings-emnlp.462

Cheng Xu, Shuhao Guan, Derek Greene, and M. Kechadi. 2024. Benchmark data contamination of large language models: A survey. *arXiv preprint arXiv:2406.04244*.

Fanghua Ye, Jarana Manotumruksa, and Emine Yilmaz. 2022. MultiWOZ 2.4: A multi-domain task-oriented dialogue dataset with essential annotation corrections to improve state tracking evaluation. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 351–360, Edinburgh, UK. Association for Computational Linguistics.

Dian Yu, Mingqiu Wang, Yuan Cao, Izhak Shafran, Laurent Shafey, and Hagen Soltau. 2022. Unsupervised slot schema induction for task-oriented dialog. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1174–1193, Seattle, United States. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.naacl-main.86

A Schema Correction Guideline

For each task domain within a scenario, there are two schemas that must be reviewed and corrected: the Task Schema, which defines user preferences and constraints, and the Agent Knowledge Schema, which describes candidate items or solutions the agent can reason over. These two schemas should align in structure but differ in purpose. Use the following steps to ensure both schemas are complete, consistent, and clearly defined:

1. **Filter out nonsensical or inconsistent fields.** Examine the fields in both the Task Schema and the Agent Knowledge Schema. Remove any fields that are irrelevant, redundant, inconsistent, or more appropriate for a different domain within the same scenario.
2. **Check and correct field types.** Make sure all fields use data types that accurately reflect their intended use. Revise types as needed—for example, convert booleans to categorical fields containing a list of possible values when appropriate.
3. **Edit field names and descriptions for clarity and specificity.** Update field names to be clear and descriptive. Revise field descriptions to ensure they are accurate, grammatically clear, and aligned with the field type. Descriptions should help users and systems understand the role of the field in determining an optimal solution to the task without ambiguity.
4. **Add missing fields.** Identify and add important fields that are absent but necessary for realistic task performance. In the Task Schema, add fields users would expect to express preferences about. In the Agent Knowledge Schema, include fields that capture essential attributes of actual task items or solutions.
5. **Validate alignment between Task Schema and Agent Knowledge Schema.** Ensure that the Task Schema represents *user preferences, constraints, or goals*, while the Agent Knowledge Schema represents *the actual properties of candidate solutions*. Align the fields conceptually between the two, but avoid directly copying when semantics differ. Do not include preference-like constructs (e.g., ‘‘min_x’’, ‘‘max_x’’, or ‘‘preferred_x’’) in the Agent Knowledge Schema, and avoid concrete values in the Task Schema unless expressing a user constraint.
6. **Final Scenario Check.** After revising both schemas for each domain in a scenario, conduct a final review of all schemas to ensure they form a coherent task representation for the scenario as a whole. Ensure that the progression of domain tasks within the scenario reflects a plausible dialogue application setting, making further edits to each schema's fields as needed to improve the coherence of each schema within the broader dialogue scenario.

B Dialogue Correction Guideline

Each scenario in the DOTS dataset contains dialogues that must be reviewed and corrected to ensure they are accurate, natural, and coherent. Use the following steps to correct each dialogue:

1. **Preserve task structure and speaker roles.** Confirm that the dialogue maintains an appropriate alternation of speaker turns and reflects the intended task flow (e.g., goal elicitation, clarification, recommendation). The assistant should provide helpful, relevant responses; the user should express realistic goals or preferences. If the speaker roles or task adherence is problematic, make a minimal rewrite of the dialogue content to resolve the issues while keeping the content as close to the original.
2. **Ensure natural and fluent dialogue turns.** Review each utterance for fluency, grammaticality, and conversational naturalness. Revise any phrasing that is awkward, robotic, or repetitive. Pay special attention to transitions between task domains, revising dialogue turns as needed to make these transitions natural and coherent.
3. **Verify factual correctness of slot values.** Read through each dialogue and identify any slot values that are incorrect, missing, or hallucinated. Replace them with accurate values that are consistent with the underlying schema and the rest of the dialogue context.

4. **Clarify vague or ambiguous values.** Edit any slot values that are overly vague, underspecified, or ambiguous. Ensure that each value is as informative and specific as possible without changing the intended meaning. Use domain-appropriate terminology.

C Slot Mapping Procedure Guideline

For each predicted slot:

1. Review the 5 value examples randomly sampled from each predicted slot, each shown with its surrounding dialogue context.
2. Compare each sampled value to the descriptions of slots in the gold schema. Identify candidate slots that semantically match.
3. Validate a match only if at least 4 out of 5 examples align clearly with the meaning of a single gold slot. Discard ambiguous candidates.
4. Use the predicted slot name and description only for disambiguation in cases where the values are too generic (e.g., ‘‘True’’, ‘‘None’’) to determine whether a mapping is appropriate. These metadata should not drive the mapping unless value example interpretation alone is inconclusive.
5. Label the predicted slot with the single gold slot that it best matches, if such a match exists. If there is no suitable gold slot to match to a predicted slot, that predicted slot is assigned no mapping.