Zero-Shot Generalizable End-to-End Task-Oriented Dialog System using Context Summarization and Domain Schema

Adib Mosharrof [1], **M.H. Maqbool** [2], **A.B. Siddique** [1]
adib.mosharrof@uky.edu, hasanmaqbool@knights.ucf.edu, siddique@cs.uky.edu
[1]University of Kentucky, [2]University of Central Florida

Abstract

Task-oriented dialog systems empower users to accomplish their goals by facilitating intuitive and expressive natural language interactions. State-of-the-art approaches in task-oriented dialog systems formulate the problem as a conditional sequence generation task and fine-tune pre-trained causal language models in the supervised setting. This requires labeled training data for each new domain or task, and acquiring such data is prohibitively laborious and expensive, thus making it a bottleneck for scaling systems to a wide range of domains. To overcome this challenge, we introduce a novel **Z**ero-**S**hot generalizable end-to-end **T**ask-**o**riented **D**ialog system, ZS-ToD, that leverages domain schemas to allow for robust generalization to unseen domains and exploits effective summarization of the dialog history. We employ GPT-2 as a backbone model and introduce a two-step training process where the goal of the first step is to learn the general structure of the dialog data and the second step optimizes the response generation as well as intermediate outputs, such as dialog state and system actions. As opposed to state-of-the-art systems that are trained to fulfill certain intents in the given domains and memorize task-specific conversational patterns, ZS-ToD learns generic task-completion skills by comprehending domain semantics via domain schemas and generalizing to unseen domains seamlessly. We conduct an extensive experimental evaluation on SGD and SGD-X datasets that span up to 20 unique domains and ZS-ToD outperforms state-of-the-art systems on key metrics, with an improvement of **+17% on joint goal accuracy** and **+5 on inform**. Additionally, we present a detailed ablation study to demonstrate the effectiveness of the proposed components and training mechanism.

1 Introductions

Task-Oriented Dialog (ToD) systems facilitate users to achieve their goals through the utilization of human-like language interactions. Traditionally, ToD systems have employed diverse modular architectures (Molich and Nielsen 1990), incorporating separate components for Natural Language Understanding (NLU) (Mairesse et al. 2009; Lee et al. 2019), Dialogue State Tracking (DST) (Ren et al. 2018;

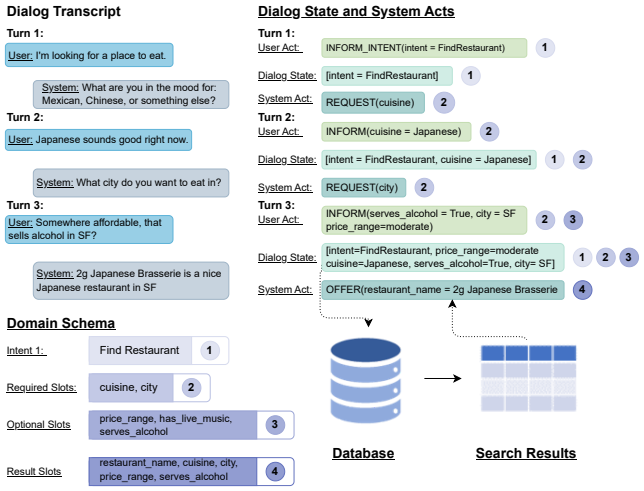

Figure 1: Overview of ZS-ToD: the domain schema facilitates estimating dialog state, system actions, and system response irrespective of whether the model was trained on that domain or not. Parts of the schema that assist in the generation are grouped by similar colors.

Lee 2013), Dialogue Policy (POL) (Peng et al. 2018; Le et al. 2021b; 2021a), and Natural Language Generation (NLG) (Wen et al. 2015; Peng et al. 2020) that are connected in a pipeline. Other variations of the pipeline also exist where NLU and DST are merged into a single module, named Word-DST (Ramadan, Budzianowski, and Gasic 2018), while POL and NLG are integrated into a single module, called Word-POL (Chen et al. 2019; Budzianowski et al. 2018a). Moreover, End-to-End (E2E) systems have emerged, producing a natural language response directly from the user's input without using intermediate stages (Bordes, Boureau, and Weston 2016). E2E ToD systems that include intermediate outputs (e.g., DST, POL) have gained increasing popularity recently since such systems can utilize intermediate outputs to facilitate effective communication with external APIs (Zhang, Ou, and Yu 2020).

The state-of-the-art (SOTA) approaches in ToD systems formulate the problem as a conditional sequence generation task and finetune pre-trained causal language models in a supervised manner using single-domain or multi-domain

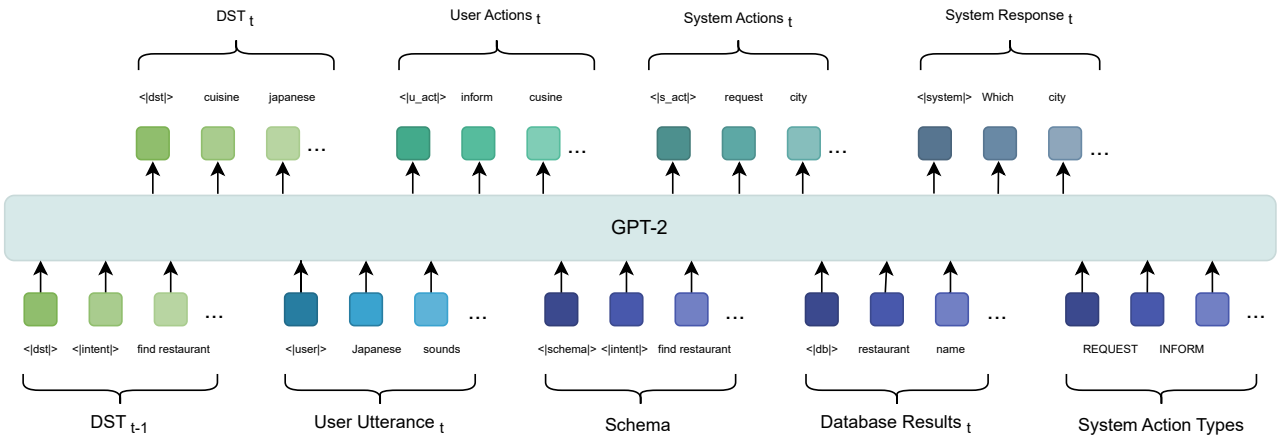

Figure 2: Overview of our approach. A GPT-2 model is fed the dialog state of the previous turn, the last user utterance, relevant schemas, database search results, and a list of system action names. As output, the model autoregressively generates the current dialog state, user actions, system actions, and system response.

datasets (Hosseini-Asl et al. 2020; Peng et al. 2021; Lee et al. 2020; Yang, Li, and Quan 2020; Jeon and Lee 2021; Sun et al. 2022; Yang et al. 2022a). These systems feed the dialog history to the model as input and the output is a cascaded generation (Su et al. 2021a) of the DST, POL, and NLG. Such systems are typically trained using large amounts of labeled data. In order for a dialog system to perform well on a specific task or in a specific domain, it needs to be trained on a large amount of labeled data that is specific to that task or domain. A major drawback of most of these systems is that they fail to generalize to unseen domains and acquiring data for each new domain is prohibitively laborious and expensive, which motivates zero-shot generalizable ToD systems. Recently, researchers have shown the possibility of building zero-shot generalizable individual components like the DST, next action prediction, and response generation, for ToD systems (Lee, Cheng, and Ostendorf 2021; Siddique, Jamour, and Hristidis 2021; Mehri and Eskénazi 2021; Siddique et al. 2021). Nonetheless, to the best of our knowledge there has been no work on building zero-shot generalizable end-to-end ToD systems.

We propose a novel **Z**ero-**S**hot generalizable end-to-end **ToD** system, ZS-ToD, using context summarization and domain schema. The domain information can be represented in the form of a schema, which contains a set of intents and the relevant set of slots needed to fulfill a given intent. The domain schema can facilitate zero-shot generalization to unseen domains in ToD systems. Figure 1 shows a few example turns from a dialog, the intermediate outputs consisting of the DST, user actions, system actions, and the domain schema for that dialog. In the first turn, the schema guides the system to infer the user intent, and the first few system actions are based on the required slots of the schema. As the conversation continues, the user makes additional queries that are part of the optional slots. Similarly, when the system queries the database using the query parameters from the DST, the results contain the slot names listed in the result slots. The top search result is usually offered to the user and if the user rejects the offered item, the next option from the search results is offered.

SOTA ToD systems use dialog history as the context to generate a response for a given turn. Generally, the dialog history consist of multiple turns (e.g., more than 20 turns) often containing conflicting information. For example, a slot value can be updated in later turns or the active intent of the user could change. At each turn, the system needs to model long as well as short range dependencies in the dialog context to accurately predict the current dialog state, and any errors made at any step would propagate to future steps. To alleviate this problem, we propose to replace the dialog history with the dialog state from the previous turn, as this would provide a concise summary of all the previous turns and allow the system to focus on the current state rather than previous states. Using a summarized context reduces the context size by a significant amount, thus allowing us to feed extra information, such as domain schema, database results and list of system action types, which otherwise would not have been possible without using a larger language model.

The goal of ToD systems is to generate a response (and intermediate outputs if needed), so there should be an explicit focus on the loss calculation for the response. To fulfill this requirement, we propose a two-step training process, where the first step focuses on understanding the structure of the dialog data, and the second step focuses on generating the correct response. SOTA ToD models are passed an input prompt (i.e, dialog history) and generate a response for the given prompt. However, these systems are trained using a cross-entropy loss over the whole sequence, i.e, dialog history and response. This kind of loss calculation for optimization might give the model superfluous rewards for correctly predicting the input prompt.

To evaluate the effectiveness of our proposed model, we conducted extensive evaluations using Schema Guided Dialogue (SGD) and SGD-X dataset that span multi-domain dialogs across 20 domains. ZS-ToD outperforms existing baseline systems across key metrics, particularly with a **+17% joint goal accuracy** and **+5 inform** improvement over prior work, demonstrating the feasibility of our approach for Zero-Shot generalizable end-to-end ToD systems.

2 Methodology

2.1 Pre-trained Language Models

Language models (e.g., BERT (Devlin et al. 2019), GPT-2 (Radford et al. 2019)) have been trained on massive amounts of textual data and have shown state-of-the-art results in a variety of NLP tasks. In this work, we use GPT-2 as the base model and fine-tune it on task-oriented dialog data to build a zero-shot generalizable end-to-end ToD system. The GPT-2 model is pre-trained for autoregressive generation (i.e., predicting the next token given past tokens) on the WebText dataset (i.e., 40 GB of textual data) and adapts a transformer-based neural architecture (Vaswani et al. 2017). Suppose we have a natural language sequence $(s_1, \cdots, s_n)$ where symbol s_i is drawn from a fixed set of symbols. The sequential ordering of language leads to factorizing the joint probabilities over symbols as a product of conditional probabilities (Bengio et al. 2003), as given below.

$$p(s) = \prod_{i=1}^{n} p(s_i|s_1, \cdots, s_{i-1})$$

Using this approach, it is possible to estimate $p(s)$ and any conditionals of the form $p(s_{i-k}, \cdots, s_i|s_1, \cdots, s_{i-k-1})$, and perform tractable sampling. Since we formulate our problem as a sequence generation problem, GPT-2 is a natural choice for our TOD system.

2.2 Problem Formulation

We formulate task-oriented dialog response generation as a conditional sequence generation task. To facilitate zero-shot generalization to unseen domains, we condition the response generation for a dialog x on the domain schema S_i, in addition to the dialog context. The overall loss of our model can be written as:

$$\mathcal{L}_{ZS-ToD} = \mathbb{E}_{x\sim D}\left(\sum_{n=1}^{T} -\log p(x_n|x_{<n}, S_i)\right) \quad (1)$$

Specifically, in a multi-domain dialog system, the domain semantics are encapsulated in the domain schema denoted by $S = \{S_1, S_2, \cdots\}$ where S_i represents schema for domain D_i, which consists of a set of intents $I = \{I_1, I_2, \cdots\}$ and relevant set of slots $K = \{K_1, K_2, \cdots\}$ to fulfill a given intent. A dialog session is composed of interactions between the user and the system in natural language from (single or) multiple domains. We denote the dialog as $\{U_1^u, U_1^s, \cdots, U_T^s\}$ where U_i^u and U_i^s represents user and system utterances, respectively, at turn i and T is the number of turns in a dialog. We summarize the dialog history up to turn $t-1$ in the form of a dialog state represented by DS_{t-1}, which tracks the user's active intent I_k and a list of tuples recording the slot names and corresponding slot values in a particular domain (D_j, K_i, V_i), where V_i represents the value for the slot K_i. At turn t, ZS-ToD estimates the probability of the dialog state DS_t by conditioning on previous dialog state DS_{t-1}, user's current utterance U_t^u, and domain schema S_i:

$$P(DS_t|DS_{t-1}, U_t^u, S_i) \quad (2)$$

Then, at turn t, ZS-ToD estimates the probability of the user action A_t^u by conditioning on dialog state DS_t, user's current utterance U_t^u, and domain schema S_i:

$$P(A_t^u|DS_t, U_t^u, S_i) \quad (3)$$

where the user action A_t^u is represented by a list of tuples (D_j, a_n^u, K_i, V_i) to record the user's action type $a_i^u \in \{a_1^u, a_2^u, \cdots, a_m^u\}$, slot name K_i, and corresponding slot values V_i in a particular domain D_j. The estimated dialog state DS_t at turn t is used to query the database (if needed), which returns a list of database results denoted by DB_t, that satisfy the constraints in the dialog state.

Next, ZS-ToD estimates the probability of the system action A_t^s denoted by a list of tuples (D_j, a_m^s, K_i, V_i) where $a_i^s \in \{a_1^s, a_2^s, \cdots, a_n^s\}$ represents system's action type by conditioning on the dialog state DS_t, user utterance U_t^u, user action A_t^u, database results DB_t, the set of all available system action types $\forall a_i^s$:

$$P(A_t^s|DS_t, U_t^u, A_t^u, DB_t, \forall a_i^s) \quad (4)$$

Finally, ZS-ToD estimates the probability of the system's natural language response U_t^s by conditioning on the dialog state DS_t, user utterance U_t^u, user action A_t^u, system action A_t^s, and domain schema S_i:

$$P(U_t^s|DS_t, U_t^u, A_t^u, A_t^s, S_i) \quad (5)$$

To accommodate for multi-domain dialogs, multiple domain schemas can be used to condition on. In the traditional supervised learning setting, we have labeled training dialogs from all the domains. Whereas, in the zero-shot learning setup we assume that the training dialogs are available only for seen domains $D_s = \{D_1, D_2, \cdots, D_k\}$ and the dialogs from unseen domains $D_u = \{D_{k+1}, D_{k+2}, \cdots\}$ may only show up at inference time where $D_s \cap D_u = \emptyset$. This challenging setting is the focus of the paper.

2.3 Model Architecture

We use the pre-trained GPT-2 with a language modeling (LM) head. Following the autoregressive nature of the model and problem formulation, we feed the previous dialog state, the user's current utterance, domain schema, and optionally database results to ZS-ToD after tokenization. Then, ZS-ToD outputs a representation, which upon decoding represents the updated dialog state, user actions, system actions, and system response in natural language. In this work, we employ the greedy decoding strategy for text generation.

2.4 Two-step Training

Following the conditional generation in GPT-2, at a given turn t, we pass input tokens $S_{in} = \{s_1, \cdots, s_k\}$ as the prompt and the model generates a response $S_{out} = \{s_{k+1}, \cdots, s_{m+k}\}$, where k and m represent the input and output lengths, respectively. Since conditional generation models (e.g., GPT-2) process one token at a time, the probability of predicting a token s_i can be written as: $p(s_i|s_1, \cdots, s_{i-1})$. The standard procedure for training these models is to optimize the Cross-Entropy (CE) loss over the full sequence. In ToD systems, the input prompt

Model	Domains	Intent Accuracy	Requested Slots F1	Average Goal Accurracy	Joint Goal Accuracy	Inform	Success	Average Action Accuracy	Joint Action Accuracy	Average UserAction Accuracy	Joint UserAction Accuracy	Response GLEU	Combined
SimpleTOD	all	78.60	94.08	47.85	24.18	55.65	47.27	49.08	37.66	66.42	57.46	20.64	72.10
	seen	80.07	94.55	52.00	29.35	58.35	50.13	51.43	40.26	68.88	60.31	24.89	79.13
	unseen	78.63	93.92	46.27	22.72	54.28	46.17	48.29	37.12	65.55	56.65	19.24	69.47
SimpleTOD w/ Schema & DB Results	all	82.34	95.72	58.03	30.36	68.30	60.47	55.18	43.42	70.30	60.23	22.03	86.41
	seen	83.32	96.05	61.29	34.88	70.05	62.68	57.28	46.01	72.34	62.61	25.68	92.04
	unseen	82.19	95.71	57.35	29.20	68.10	60.48	54.64	42.85	70.19	60.24	20.40	84.69
ZS-ToD	all	**84.83**	**95.53**	**72.38**	**48.44**	**73.08**	**62.19**	**58.32**	**46.31**	**73.20**	**64.20**	**20.04**	**87.67**
	seen	**85.48**	**95.88**	**74.23**	**52.05**	**74.72**	**63.85**	**60.19**	**48.69**	**74.89**	**66.24**	**24.66**	**93.95**
	unseen	**84.45**	**95.42**	**72.03**	**47.83**	**71.68**	**61.63**	**57.42**	**45.21**	**72.56**	**63.46**	**18.51**	**85.16**

Table 1: Main Results. For end-to-end systems, ZS-ToD outperforms existing baselines across all metrics, particularly there is significant improvement in key metrics like Average/Joint Goal Accuracy and Inform.

is usually a long sequence of text that contains the entire dialog history, and the generation is much shorter than the input prompt. Since the focus is not on generating the input prompt, we need to modify the loss function to pay less attention to the input prompt and more attention to the response.

To overcome the aforementioned issue, we propose a two step training approach. In the first step, we follow the standard training procedure and calculate the CE loss on the full sequence. For the second step, we initialize the model with the weights from the first step and calculate the CE loss only on the response, as shown in Equation (6).

$$L = -\sum_{i=k+1}^{k+m} s_i \log(p_i) \tag{6}$$

2.5 Zero-shot Generalization

Once ZS-ToD is trained using the above-mentioned techniques, it can generalize to unseen domains seamlessly. When ZS-ToD is exposed to dialogs from a new unseen domain, the domain schema is expected to be part of the input. Since the problem is formulated as a conditional generation, ZS-ToD pays attention to the relevant parts of the schema to generate the user intent, slot names as well as slot values, thus adapting to the new unseen domains with no additional training.

3 Experimental Setup

3.1 Datasets

The Schema Guided Dialogue (SGD). SGD dataset is a large-scale dataset for task-oriented dialogue that consists of over 16K multi-domain dialogs between a human and a virtual assistant covering 20 domains. The dataset also provides a schema for each domain that provides a textual description of the domain, a list of slots, and a list of intents. A slot contains a name, textual description, and possible values for categorical slots and an intent contains a name, textual description, optional slots, and result slots.

SGD-X. SGD-X dataset is an extension of the SGD dataset that contains stylistic variants for every schema in SGD. It provides 5 variants of domain schemas, where each variant incrementally moves further away from the original schema. The goal of this dataset is to evaluate model sensitivity to schema variations. The authors of the dataset have shown that two of the top-performing schema-guided DST models are sensitive to schema changes and have had significant performance drops on SGD-X.

3.2 Evaluation Metrics

To evaluate the performance of our model, we compute multiple metrics on each component of a ToD system.

DST. We evaluate the performance of DST by calculating the intent accuracy, average goal accuracy (AGA), joint goal accuracy (JGA), and requested slot F1.

System Actions. To evaluate the system actions, we compute the following metrics: inform, success, average action accuracy (AAA), and joint action accuracy (JAA). Inform measures whether a system has provided a correct entity and success measures whether it has answered all the requested information. AAA and JAA are similar to the goal metrics and are calculated from system actions. For inform, from the ground truth system actions we filter actions by action type inform (Inform, Inform Count) and check if they are predicted correctly. For success, we filter actions by slot names that are in the requested slots and check if the action slot values are predicted correctly. AAA and JAA are implemented following the implementations of AGA and JGA. Since we also predict user actions, we calculate the average and joint accuracy of the predicted user actions.

System Response. For evaluating the system response, we report the GLEU (Wu et al. 2016) score as it performs better on individual sentence pairs.

Overall. To get an overall score for the model, we calculate the combined score (Mehri, Srinivasan, and Eskenazi 2019): (Inform + Success) $\times$ 0.5 + GLEU.

To ensure a fair comparison of ZS-ToD with existing systems that have reported results on the SGD dataset, we use the evaluation script provided by the SGD dataset, where applicable.

4 Results

Main Results. Since no E2E ToD system has reported results for the SGD dataset, we follow (Hosseini-Asl et al. 2020) to implement some of the popular baseline methods to compare with our approach and present the results in Table 1. We can see that ZS-ToD outperforms all the baselines across all metrics except GLEU, where its performance is super competitive (e.g., 24.89 vs 24.66). An explanation of this could be that since we replaced the dialog history with the dialog state, the performance of the model improved on all other metrics, but the model lost a lot of exposure to dialog utterances. Another reason could be greedy decoding which works well for a structured generation but is not the best strategy for fluent text generation. While the system response requires a fluent generation, all other parts of the

Model	Domains	Intent Accuracy	Requested Slots F1	Average Goal Accurracy	Joint Goal Accuracy	Inform	Success	Average Action Accuracy	Joint Action Accuracy	Response GLEU	Combined
ZS-ToD **(this work)**	all	**84.83**	**95.53**	**72.38**	**48.44**	**73.08**	**62.19**	**58.32**	**46.31**	**20.04**	**87.67**
	seen	**85.48**	**95.88**	**74.23**	**52.05**	**74.72**	**63.85**	**60.19**	**48.69**	**24.66**	**93.95**
	unseen	**84.45**	**95.42**	**72.03**	**47.83**	**71.68**	**61.63**	**57.42**	**45.21**	**18.51**	**85.16**
w/o Two Step Training	all	75.08	92.80	62.47	39.52	48.13	44.27	40.38	30.71	11.41	57.61
	seen	75.75	93.13	64.66	42.76	50.26	46.47	41.96	32.42	13.75	62.11
	unseen	75.79	92.90	62.60	39.25	47.55	44.32	40.25	30.66	11.03	56.97
w/o Domain Schema	all	82.14	94.67	64.70	38.47	59.88	53.88	54.14	43.07	21.15	78.03
	seen	83.34	95.10	67.62	43.39	62.30	56.64	56.61	45.92	27.10	86.57
	unseen	81.96	94.52	63.95	37.59	58.65	53.25	53.22	42.20	19.33	75.28
w/o DB Results	all	82.50	95.48	71.54	43.20	50.96	56.89	53.67	41.73	17.62	71.54
	seen	83.26	95.85	73.87	47.62	53.03	59.08	55.73	43.91	23.12	79.17
	unseen	82.19	95.36	71.04	42.17	50.33	56.95	53.17	41.52	16.07	69.70
w/o Sys Action Names	all	82.56	96.00	72.86	44.52	60.13	61.91	57.98	45.86	21.02	82.04
	seen	83.25	96.32	75.11	48.77	61.69	64.04	60.12	48.37	26.56	89.43
	unseen	82.38	95.91	72.44	43.60	59.61	61.75	57.29	45.26	19.16	79.84

Table 2: Ablation Study of ZS-ToD.

Model	Intent Accuracy	Requested Slot F1	Average GA	Joint GA
SGD Baseline	90.60	**96.50**	56	25.40
FastSGT	90.33	96.33	60.66	29.20
Seq2Seq-DU	**91.00**	-	-	30.10
DSGFNET	-	-	-	32.10
ZS-ToD	81.49	95.97	**74.08**	**49.73**

Table 3: Results on SGD test set. Our approach significantly outperforms baselines methods in terms of average and joint goal accuracy.

generation can be deemed as a structured generation. On the other hand, nucleus and top-k sampling strategies are better suited for a fluent generation but are not the best for a structured generation. We formulated the problem as a single sequence generation, and we can only select one strategy, so there is bound to be a trade-off. Since there is no single strategy that is best suited for both fluent and structured generation, our selection of greedy decoding may have been the cause for the loss of fluency in response generation.

We evaluate the DST performance of ZS-ToD with the evaluation script provided by the SGD dataset and present our results alongside other baseline DST models in Table 3. We can see that even though our method is not specifically designed for DST, still it significantly outperforms the baselines models in the important metrics: Average and Joint Goal Accuracy.

Long Range Dialog Dependencies. In order to process dialogs that have a large number of turns, a system must be effective at capturing long-range dependencies. To test this ability, we group the test dialogs based on the number of turns and evaluate on each group. As shown in Figure 3, ZS-ToD outperforms the baseline systems across all groups. Generally, in the first few turns of a dialog, the main focus is on figuring out what the user wants. The user could switch among mulitple options before finally deciding on one, however towards the end of a dialog, usually the user has a clear idea of what he or she wants, so is less likely to make many changes. For the first few turns, we have observed that there is a steeper drop in performance of the baseline when compared to ZS-ToD . A possible explanation of this could be that, since we pass the dialog summary to the model, it contains the correct state of the dialog at the previous turn, which helps the model to make better predictions. In groups with large number of turns, both the baseline and ZS-ToD perform similarly, which suggests even though ZS-ToD does well in capturing medium range dependencies, long range dependencies are still a challenge.

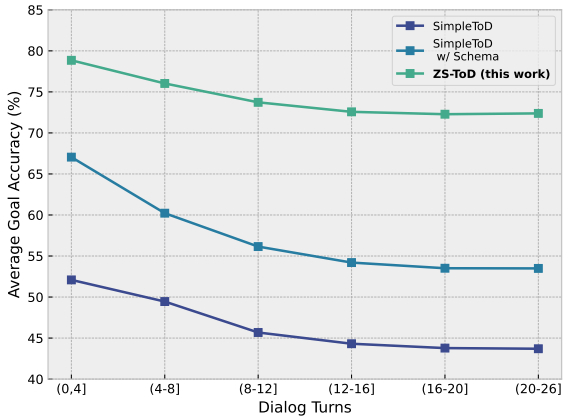

Figure 3: Performance of dialog systems on the SGD test set with respect to dialog turns

Two Step Training. To better understand the effect of the two step training process, we compared ZS-ToD and a few baseline systems with and without the two step training process. In Figure 4, we can see that models that incorporate schema benefit from the two step training process.

Ablation Study. To get a better understanding of the different components of our model, we drop a certain component of ZS-ToD to show the effect on the performance and report an ablation study in Table 2. We can see that dropping two step training drastically degrades performance across all metrics, which suggests the importance of the training mechanism for ZS-ToD.

The role of schema is also important as we can see that the performance of ZS-ToD drops across all metrics when we drop schema. Another important aspect to notice here is that this variant has the largest difference in performance between seen and unseen domains. These observations indicate that schema not only aids the model to generalize to

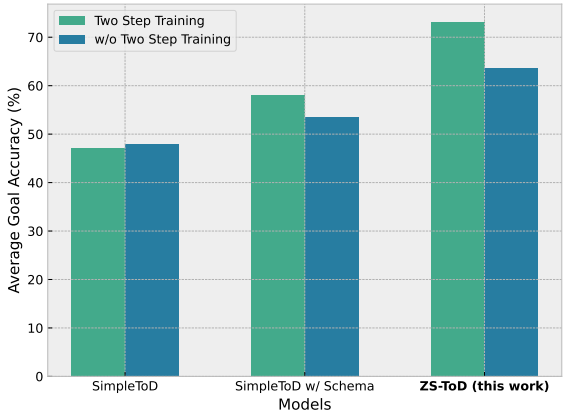

Figure 4: Effect of Two Step Training on dialog systems

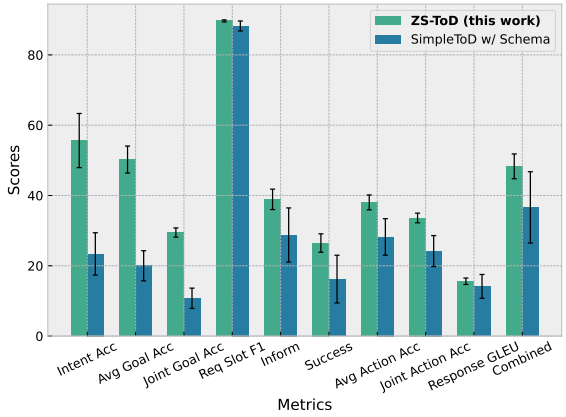

Figure 5: SGD-X results: Mean and standard deviation of each metric across all versions of SGD-X

new domains, but also plays a central role in the overall performance of the system. When the database results are excluded from the input, there is a big drop in metrics related to system actions. Additionally, there is a small drop in the DST performance as well, which suggests that there is some correlation between the database results and DST. When we omit the list of system actions types, the metrics related to system actions decreases the most, particularly Inform, but the drop in performance is much less when compared to the setting when the database results were dropped. However, in this setting there were no changes to metrics related to DST.

Results on SGD-X. To access the robustness of ZS-ToD, we ran experiments on the unseen domains of the SGD-X dataset and present the results in Figure 5. The bar graph shows the mean of each metric across all the versions of SGD-X and the error bars show the standard deviation. ZS-ToD outperforms the baseline across all metrics and has lower standard deviation, showing the robustness of ZS-ToD to domain schema variations.

5 Related Works

Supervised End to End Models. Pretrained language models like BERT (Devlin et al. 2019), GPT-2 (Radford et al. 2019), T5 (Raffel et al. 2019) and UniLM (Dong et al. 2019) have been used extensively in the literature for end-to-end models for ToD systems (Hosseini-Asl et al. 2020; Peng et al. 2021; Lee et al. 2020; Yang, Li, and Quan 2020; Jeon and Lee 2021; Sun et al. 2022; Yang et al. 2022a; Noroozi et al. 2020; He et al. 2021) on the popular MultiWoz 2.0 (Budzianowski et al. 2018b) dataset. Even though some of these models has shown experiments on zero shot performance, they shine under supervised settings and are not able to generalize to new domains, whereas our model is designed to be zero shot generalizable. In all the existing systems, the dialog history is passed as the context, whereas we use a summarized context which consists of the current user utterance and the DST of the previous state as the context.

Schema Guided Models. To obtain Zero Shot generalizability, some work has been done by incorporating schema to transfer knowledge across domains. However these systems only focus on certain components of ToD systems, such as DST (Feng, Wang, and Li 2020; Feng et al. 2022; Lee, Cheng, and Ostendorf 2021; Noroozi et al. 2020; Wang et al. 2022) and next action prediction and response generation (Mosig, Mehri, and Kober 2020; Mehri and Eskénazi 2021).

Description and Prompt Based Models. Generally, schema is described using abbreviated notations or in snake case, and this vocabulary that is not usually present in natural language. To remedy this problem, there has been some work on description based DST (Zhao et al. 2022; Lin et al. 2021b; Mi et al. 2021), where the abbreviated and unnatural words are converted into natural descriptions, from which models can obtain useful semantic descriptions. Another aspect of ToD systems is slot filling, which has been formulated as a question answering problem (Yang et al. 2022b; Madotto et al. 2021; Hu et al. 2022; Brown et al. 2020; Su et al. 2021b; Lin et al. 2021a; Li et al. 2021), where a prompt is passed as a natural language question and the model predicts the slot value. These models are making the context larger and require large language models to fit such a big input, whereas we embark on the opposite direction and try to make the context smaller.

6 Conclusion

We have presented a novel schema-guided zero-shot generalizable end-to-end task-oriented dialog system that estimates a concise summary of the dialog history through the dialog state. This system leverages domain schemas and effective dialog history summarization to allow for robust generalization to unseen domains, overcoming the major bottleneck of acquiring labeled training data for each new domain. Additionally, a two-stage training methodology was introduced, wherein the model initially acquires an understanding of the general structure of the dialog data and subsequently optimizes the response generation process. The experimental results show the superiority of our proposed ZS-ToD over state-of-the-art models on key metrics, particularly with a **+17% joint goal accuracy** and **+5 inform** improvement over prior work. To better evaluate the effectiveness of the proposed components and training mechanisms, we have provided an ablation study that shows the significance of our contributions.

References

Bengio, Y.; Ducharme, R.; Vincent, P.; and Jauvin, C. 2003. A neural probabilistic language model. *Journal of Machine Learning Research* 3:1137–1155.

Bordes, A.; Boureau, Y.-L.; and Weston, J. 2016. Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683*.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T. J.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language models are few-shot learners. *ArXiv* abs/2005.14165.

Budzianowski, P.; Casanueva, I.; Tseng, B.-H.; and Gasic, M. 2018a. Towards end-to-end multi-domain dialogue modelling.

Budzianowski, P.; Wen, T.-H.; Tseng, B.-H.; Casanueva, I.; Stefan, U.; Osman, R.; and Gašić, M. 2018b. 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 (EMNLP)*.

Chen, W.; Chen, J.; Qin, P.; Yan, X.; and Wang, W. Y. 2019. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. *arXiv preprint arXiv:1905.12866*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv* abs/1810.04805.

Dong, L.; Yang, N.; Wang, W.; Wei, F.; Liu, X.; Wang, Y.; Gao, J.; Zhou, M.; and Hon, H.-W. 2019. Unified language model pre-training for natural language understanding and generation. *ArXiv* abs/1905.03197.

Feng, Y.; Lipani, A.; Ye, F.; Zhang, Q.; and Yilmaz, E. 2022. Dynamic schema graph fusion network for multi-domain dialogue state tracking. *ArXiv* abs/2204.06677.

Feng, Y.; Wang, Y.; and Li, H. 2020. A sequence-to-sequence approach to dialogue state tracking. In *Annual Meeting of the Association for Computational Linguistics*.

He, W.; Dai, Y.; Zheng, Y.; Wu, Y.; Cao, Z.; Liu, D.; Jiang, P.; Yang, M.; Huang, F.; Si, L.; Sun, J.; and Li, Y. 2021. Galaxy: A generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection. In *AAAI Conference on Artificial Intelligence*.

Hosseini-Asl, E.; McCann, B.; Wu, C.-S.; Yavuz, S.; and Socher, R. 2020. A simple language model for task-oriented dialogue. *ArXiv* abs/2005.00796.

Hu, Y.; Lee, C.-H.; Xie, T.; Yu, T.; Smith, N. A.; and Ostendorf, M. 2022. In-context learning for few-shot dialogue state tracking. In *Conference on Empirical Methods in Natural Language Processing*.

Jeon, H., and Lee, G. G. 2021. Dora: Toward policy optimization for task-oriented dialogue system with efficient context. *Comput. Speech Lang.* 72:101310.

Le, N.; Siddique, A.; Jamour, F.; Oymak, S.; and Hristidis, V. 2021a. Generating predictable and adaptive dialog policies in single-and multi-domain goal-oriented dialog systems. *International Journal of Semantic Computing* 15(04):419–439.

Le, N.; Siddique, A.; Jamour, F.; Oymak, S.; and Hristidis, V. 2021b. Predictable and adaptive goal-oriented dialog policy generation. In *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*, 40–47. IEEE.

Lee, S.; Zhu, Q.; Takanobu, R.; Li, X.; Zhang, Y.; Zhang, Z.; Li, J.; Peng, B.; Li, X.; Huang, M.; et al. 2019. Convlab: Multi-domain end-to-end dialog system platform. *arXiv preprint arXiv:1904.08637*.

Lee, H.; Jo, S.; Kim, H.; Jung, S.; and Kim, T.-Y. 2020. Sumbt+larl: End-to-end neural task-oriented dialog system with reinforcement learning. *ArXiv* abs/2009.10447.

Lee, C.-H.; Cheng, H.; and Ostendorf, M. 2021. Dialogue state tracking with a language model using schema-driven prompting. In *Conference on Empirical Methods in Natural Language Processing*.

Lee, S. 2013. Structured discriminative model for dialog state tracking. In *Proceedings of the SIGDIAL 2013 Conference*, 442–451.

Li, S.; Cao, J.; Sridhar, M.; Zhu, H.; Li, S.-W.; Hamza, W.; and McAuley, J. 2021. Zero-shot generalization in dialog state tracking through generative question answering. In *Conference of the European Chapter of the Association for Computational Linguistics*.

Lin, Z.; Liu, B.; Madotto, A.; Moon, S.; Crook, P. A.; Zhou, Z.; Wang, Z.; Yu, Z.; Cho, E.; Subba, R.; and Fung, P. 2021a. Zero-shot dialogue state tracking via cross-task transfer. *ArXiv* abs/2109.04655.

Lin, Z.; Liu, B.; Moon, S.; Crook, P. A.; Zhou, Z.; Wang, Z.; Yu, Z.; Madotto, A.; Cho, E.; and Subba, R. 2021b. Leveraging slot descriptions for zero-shot cross-domain dialogue statetracking. In *North American Chapter of the Association for Computational Linguistics*.

Madotto, A.; Lin, Z.; Winata, G. I.; and Fung, P. 2021. Few-shot bot: Prompt-based learning for dialogue systems. *ArXiv* abs/2110.08118.

Mairesse, F.; Gasic, M.; Jurcicek, F.; Keizer, S.; Thomson, B.; Yu, K.; and Young, S. 2009. Spoken language understanding from unaligned data using discriminative classification models. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 4749–4752. IEEE.

Mehri, S., and Eskénazi, M. 2021. Schema-guided paradigm for zero-shot dialog. In *SIGDIAL Conferences*.

Mehri, S.; Srinivasan, T.; and Eskenazi, M. 2019. Structured fusion networks for dialog. *arXiv preprint arXiv:1907.10016*.

Mi, F.; Li, Y.; Wang, Y.; Jiang, X.; and Liu, Q. 2021. Cins: Comprehensive instruction for few-shot learning in task-oriented dialog systems. In *AAAI Conference on Artificial Intelligence*.

Molich, R., and Nielsen, J. 1990. Improving a human-computer dialogue. *Communications of the ACM* 33(3):338–348.

Mosig, J. E. M.; Mehri, S.; and Kober, T. 2020. Star: A schema-guided dialog dataset for transfer learning. *ArXiv* abs/2010.11853.

Noroozi, V.; Zhang, Y.; Bakhturina, E.; and Kornuta, T. 2020. A fast and robust bert-based dialogue state tracker for schema guided dialogue dataset. *ArXiv* abs/2008.12335.

Peng, B.; Li, X.; Gao, J.; Liu, J.; Wong, K.-F.; and Su, S.-Y. 2018. Deep dyna-q: Integrating planning for task-completion dialogue policy learning. *arXiv preprint arXiv:1801.06176*.

Peng, B.; Zhu, C.; Li, C.; Li, X.; Li, J.; Zeng, M.; and Gao, J. 2020. Few-shot natural language generation for task-oriented dialog. *arXiv preprint arXiv:2002.12328*.

Peng, B.; Li, C.; Li, J.; Shayandeh, S.; Lidén, L.; and Gao, J. 2021. Soloist: Building task bots at scale with transfer learning and machine teaching. *Transactions of the Association for Computational Linguistics* 9:807–824.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners.

Raffel, C.; Shazeer, N. M.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *ArXiv* abs/1910.10683.

Ramadan, O.; Budzianowski, P.; and Gasic, M. 2018. Large-scale multi-domain belief tracking with knowledge sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume 2, 432–437.

Ren, L.; Xie, K.; Chen, L.; and Yu, K. 2018. Towards universal dialogue state tracking. *arXiv preprint arXiv:1810.09587*.

Siddique, A.; Jamour, F.; Xu, L.; and Hristidis, V. 2021. Generalized zero-shot intent detection via commonsense knowledge. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1925–1929.

Siddique, A.; Jamour, F.; and Hristidis, V. 2021. Linguistically-enriched and context-awarezero-shot slot filling. In *Proceedings of the Web Conference 2021*, 3279–3290.

Su, Y.; Shu, L.; Mansimov, E.; Gupta, A.; Cai, D.; Lai, Y.-A.; and Zhang, Y. 2021a. Multi-task pre-training for plug-and-play task-oriented dialogue system. *arXiv preprint arXiv:2109.14739*.

Su, Y.; Shu, L.; Mansimov, E.; Gupta, A.; Cai, D.; Lai, Y.-A.; and Zhang, Y. 2021b. Multi-task pre-training for plug-and-play task-oriented dialogue system. In *Annual Meeting of the Association for Computational Linguistics*.

Sun, H.; Bao, J.; Wu, Y.; and He, X. 2022. Bort: Back and denoising reconstruction for end-to-end task-oriented dialog. In *NAACL-HLT*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems* 30.

Wang, Q.; Cao, Y.; Li, P.; Fu, Y.; Lin, Z.; and Guo, L. 2022. Slot dependency modeling for zero-shot cross-domain dialogue state tracking. In *International Conference on Computational Linguistics*.

Wen, T.-H.; Gasic, M.; Mrksic, N.; Su, P.-H.; Vandyke, D.; and Young, S. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*.

Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; Klingner, J.; Shah, A.; Johnson, M.; Liu, X.; Łukasz Kaiser; Gouws, S.; Kato, Y.; Kudo, T.; Kazawa, H.; Stevens, K.; Kurian, G.; Patil, N.; Wang, W.; Young, C.; Smith, J.; Riesa, J.; Rudnick, A.; Vinyals, O.; Corrado, G.; Hughes, M.; and Dean, J. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation.

Yang, Y.; Ding, H.; Liu, Q.; and Quan, X. 2022a. Ubarv2: Towards mitigating exposure bias in task-oriented dialogs. *ArXiv* abs/2209.07239.

Yang, Y.; Lei, W.; Cao, J.; Li, J.; and Chua, T.-S. 2022b. Prompt learning for few-shot dialogue state tracking. *ArXiv* abs/2201.05780.

Yang, Y.; Li, Y.; and Quan, X. 2020. Ubar: Towards fully end-to-end task-oriented dialog systems with gpt-2. In *AAAI Conference on Artificial Intelligence*.

Zhang, Y.; Ou, Z.; and Yu, Z. 2020. Task-oriented dialog systems that consider multiple appropriate responses under the same context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 9604–9611.

Zhao, J.; Gupta, R.; Cao, Y.; Yu, D.; Wang, M.; Lee, H.; Rastogi, A.; Shafran, I.; and Wu, Y. 2022. Description-driven task-oriented dialog modeling. *ArXiv* abs/2201.08904.