# LAD: Language Models as Data for Zero-Shot Dialog

**Shikib Mehri**[♠] **Yasemin Altun**[♢] **Maxine Eskenazi**[♠]
[♠] Carnegie Mellon University [♢] Google
amehri@cs.cmu.edu,altun@google.com,max@cs.cmu.edu

## Abstract

To facilitate zero-shot generalization in task-oriented dialog, this paper proposes *Language Models as Data* (LAD). LAD is a paradigm for creating *diverse* and *accurate* synthetic data which conveys the necessary structural constraints and can be used to train a downstream neural dialog model. LAD leverages GPT-3 to induce linguistic diversity. LAD achieves significant performance gains in zero-shot settings on intent prediction (**+15%**), slot filling (**+31.4 F-1**) and next action prediction (**+11 F-1**). Furthermore, an interactive human evaluation shows that training with LAD is competitive with training on human dialogs. LAD is open-sourced, with the code and data available at **https://github.com/Shikib/lad**.

## 1 Introduction

A long-standing goal of dialog research is to develop mechanisms for flexibly adapting dialog systems to new domains and tasks (Rastogi et al., 2020; Mosig et al., 2020). While the advent of large-scale pre-training (Devlin et al., 2018; Liu et al., 2019b; Zhang et al., 2019) has brought about significant progress in few-shot and zero-shot generalization across many different problems in Natural Language Processing (Brown et al., 2020; Wei et al., 2021), zero-shot generalization in **task-oriented dialog** remains elusive. A likely reason for this discrepancy is that dialog models require significant data because they need to learn task-specific **structural constraints**, such as the domain ontology and the dialog policy. While large language models (e.g., GPT-3) exhibit strong language understanding and generation abilities (Brown et al., 2020), they have no *a priori* knowledge of the structural constraints implied by a specific (unseen) problem setting (e.g., relevant intents, dialog policy, etc.). As such, in order to adapt a pre-trained LM for task-oriented dialog, it is necessary to *impose **structural constraints** on the **unstructured** representation space of a pre-trained model*. Fine-tuning moderately-sized language models (LMs) (e.g., BERT) with well-motivated inductive biases (Mitchell, 1980) facilitates sample-efficient learning of the structural constraints (Peng et al., 2020; Henderson and Vulić, 2020; Mehri and Eskenazi, 2021b). However, fine-tuning can be impractical (e.g., in academic settings) with large LMs (e.g., GPT-3) due to the cost, computational power and immutable architectures. To this end, this paper aims to address the following: *'How can we leverage the strong language understanding and generation abilities of large LMs to facilitate **zero-shot generalization** in task-oriented dialog?'*

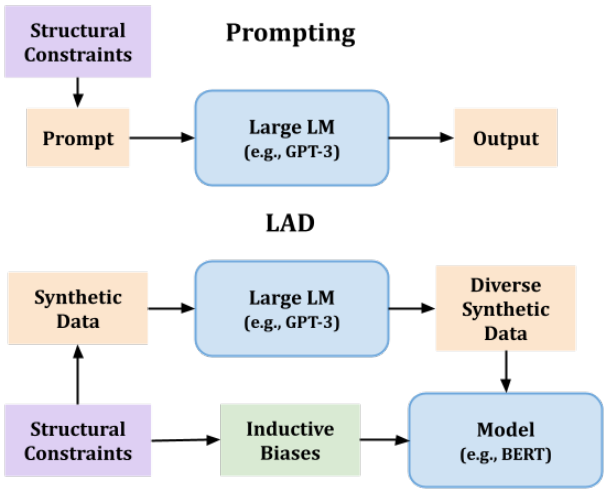


Figure 1: Prompting must convey the structural constraints through a natural language prompt. In contrast, LAD uses large LMs to induce diversity in a synthetic dataset. As such, LAD conveys structural constraints through both the synthetic data and the inductive biases in the downstream problem-specific models.

Given the in-context meta-learning abilities of large LMs (Brown et al., 2020), prior work has explored prompt-engineering or prompt-tuning (Reynolds and McDonell, 2021; Lester et al., 2021; Madotto et al., 2021). Well-designed prompts can convey the necessary structural constraints. How-



*Proceedings of the SIGdial 2022 Conference*, pages 595–604
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022 

ever, it is challenging to express complex constraints (e.g., a dialog policy) in natural language. Prompting also precludes inductive biases in the model (architecture, training algorithm, etc.) and over-relies on the meta-learning abilities of large LMs. As such, there is a tradeoff between prompting large LMs (i.e., generalizable NLU and NLG) and fine-tuning smaller LMs (i.e., problem-specific inductive biases, efficiency). A potential interpretation for the strength of large LMs is that they learn the distributional structure of language (Harris, 1954) by observing web-scale data (Sinha et al., 2021). Motivated by this interpretation, this paper proposes *Language Models as Data* (**LAD**).

**LAD** is a novel paradigm in which large LMs are used in a zero-shot domain-agnostic manner to induce *linguistic diversity* in synthetic data. Given a *minimal expression*[1] of the structural constraints (henceforth referred to as a **schema**), LAD (1) creates a seed synthetic dataset using domain-agnostic algorithms, (2) leverages large LMs to *reformulate* utterances, and (3) validates the resulting data to ensure adherence to the schema. The resulting synthetic data, which is sufficiently **diverse** and expresses the necessary **structural constraints**, can be used to train neural dialog models. In contrast to prompting, LAD facilitates zero-shot generalization by (1) leveraging the sophisticated abilities of large LMs (knowledge of the distributional structure of language) to induce *linguistic diversity* in the synthetic data while (2) maintaining inductive biases (motivated by the structural constraints) in the problem-specific model architectures.

The challenge of creating synthetic data that is indistinguishable from human-annotated data, both in its expression of structural constraints and in its diversity, is highly impractical (Lin et al., 2021; Feng et al., 2021). Instead, the goal of this work is to create synthetic data that is *sufficient* to train a sample-efficient and robust model. Therefore, the claim of this paper is that LAD can create synthetic data, conditioned on a minimal expression of structural constraints (i.e., a schema), that can be used to train robust and sample-efficient neural models and induce performance gains in zero-shot settings.

To validate this claim, LAD is applied to three problems in dialog: intent prediction, slot filling and next action prediction. Next action prediction is particularly difficult in zero-shot settings since the structural constraints include the *dialog policy*. LAD demonstrates significant gains across five datasets (**+10** to **+30** improvements on F-1 and accuracy) in zero-shot settings when evaluating on human-annotated corpora. To further validate the efficacy of LAD, an interactive evaluation with humans (over 1600 dialogs) is performed. The results of this interactive evaluation suggest that LAD can yield performance comparable to training on human dialogs. The claim of this paper is validated empirically across multiple datasets. LAD is shown to generate diverse and accurate synthetic data, which is subsequently used to train neural dialog models and facilitate zero-shot generalization.

[1]A minimal expression can be defined as the *smallest* amount of data necessary to express a structural constraint. For example, one utterance to define an intent class.

## 2 Definitions

Zero-shot generalization can be conceptualized as *imposing **structural constraints** on the unstructured representation space of a pre-trained model, using a given **schema** (i.e., minimal expression).* We begin with a neural network, $\mathcal{M}$, with general language understanding abilities and limited knowledge of task-oriented dialog (e.g., BERT (Devlin et al., 2018)). The necessary **structural constraints** that $\mathcal{M}$ must learn are implied by the target dialog setting, i.e., the problem (e.g., next action prediction), the domain (e.g., restaurants) and the task (e.g., restaurant reservation). These structural constraints conceptually define the desired properties for the representations of $\mathcal{M}$, i.e., *what must be learned* by $\mathcal{M}$. In a full-shot setting, the constraints are conveyed by a human-annotated dataset and thereby learned through supervised learning. In contrast, the goal in zero-shot generalization is to learn these structural constraints from a minimal expression, i.e., a **schema**. The following sections formally define structural constraints and schemas.

Throughout this paper, **zero-shot** refers to a setting wherein the only *human-annotated* data is the schema. Since a schema is a minimal expression of the necessary structural constraints, we argue that it is impossible to use less data, without making assumptions about the prior knowledge of a pre-trained model. Such assumptions would limit the generality of a method for zero-shot generalization.

### 2.1 Structural Constraints

To effectively adapt a model, particularly in zero-shot settings, it is imperative to define *what the model must learn*. Structural constraints conceptualize the desired properties for the representations

of a model $\mathcal{M}$. Understanding these structural constraints allows us to design an effective paradigm to facilitate zero-shot generalization. Concretely, knowledge of the structural constraints influences (1) the inductive biases (Mitchell, 1980) in the model architecture, (2) the design of the schema, and (3) the algorithms used to create synthetic data.

**Intent prediction**, for example, is the problem of classifying an utterance $u \in \mathcal{U}$ to an intent $i \in \mathcal{I}$. An intent prediction model $\mathcal{M}_I$ must learn to produce similar representations $\mathcal{M}_I(u)$ for all utterances that have the same intent. Learning this structural constraint is equivalent to transforming the unstructured representation space of $\mathcal{M}$ to the structured output space (i.e., the intent classes).

In the problem of **slot filling**, for a given utterance $u = \{w_1, w_2, \ldots, w_n\}$ and a slot key $s \in \mathcal{S}$, we must predict the corresponding slot value for $s$. The value will either be a contiguous span from $u$, $w_{i:i+k}$, or $\text{none}$. A slot filling model $\mathcal{M}_S$ must learn two sets of structural constraints. First, the representation of $u$ (or the contextual representation of $w \in u$) must follow the structural constraints of intent prediction. Second, each slot value representation $\mathcal{M}_S(w_{i:i+k})$ should be similar to other values for the slot $s$. These two constraints impose structure on both the utterance-level and the span-level representations of $\mathcal{M}_S$.

The structural constraints of intent prediction and slot filling are straightforward and are often learned by a linear layer in supervised settings (Casanueva et al., 2020; Mehri et al., 2020). The constraints for the problem of **next action prediction** are more complex. Next action prediction is the problem of predicting the next system action $a \in \mathcal{A}$ conditioned on the dialog history $u_1, u_2, \ldots, u_n$ according to some dialog policy. Given the intents and slots in the dialog history, $\mathcal{I}_D = \{i_1, i_2, \ldots, i_m\}$ and $\mathcal{S}_D = \{s_1, s_2, \ldots, s_k\}$, the dialog policy can be expressed as a function of these intents and slots, $a = \text{policy}(\mathcal{I}_D, \mathcal{S}_D)$. As such a next action prediction model $\mathcal{M}_A$ must learn (1) the structural constraints of intent prediction, (2) of slot filling and (3) the mapping defined by the $\text{policy}$ function. The complexity of third constraint led to the schema-guided paradigm (Mehri and Eskenazi, 2021b), wherein the policy is explicitly expressed rather than being learned implicitly.

### 2.2 Schema

While structural constraints conceptualize what a model $\mathcal{M}$ must learn, the schema is a minimal expression of these constraints. Imagine that our objective is to train a human (i.e., $\mathcal{M}$ with human-level language understanding and reasoning abilities) to perform task-oriented dialog. Structural constraints define *what* the human must learn. The schema is the *minimum* amount of information needed, for the human to learn the necessary structural constraints, without prior knowledge.

For **intent prediction**, we define the schema to be a single utterance $u$ for each intent $i \in \mathcal{I}$. **Slot filling** similarly relies on one utterance $u$ for each slot type $s \in \mathcal{S}$. However, this one utterance only conveys the first structural constraint of slot filling. To ensure that $\mathcal{M}_S$ can learn meaningful span-level representations, the schema for slot filling also includes multiple[2] examples of values for each slot.

**Next action prediction** has three constraints. The first two constraints are equivalent to those of intent prediction and slot filling. As such, the schema includes both (1) one utterance for each intent and (2) a set of slot values for each slot type. To express the structural constraints of the dialog policy, we leverage the graph-based representations of the task-specific dialog policy proposed by Mosig et al. (2020) and Mehri and Eskenazi (2021b).

## 3 LAD: Language Models as Data

Despite exhibiting strong language understanding and generation abilities (Brown et al., 2020), large LMs have no *a priori* knowledge of the structural constraints of task-oriented dialog. Furthermore, imposing the necessary structural constraints on large LMs is impractical due to (1) the difficulty of fine-tuning (cost, computation, immutable architectures) and (2) the limitations of natural language prompts. As such, *Language Models as Data* (**LAD**) uses GPT-3 (Brown et al., 2020) to generate **diverse** synthetic data that express the necessary task-specific **structural constraints** and can therefore be used to train neural dialog models.

**LAD** is a framework for inducing zero-shot generalization in task-oriented dialog by creating *diverse* and *accurate* synthetic data. LAD, visualized in Figure 2, is a three step process: (§3.1) domain-agnostic algorithms generate a *seed dataset* from a schema, (§3.2) GPT-3 *reformulates* utterances in

[2]While the number of slot value examples could potentially reduced to 1, up to 20 are used in this paper.

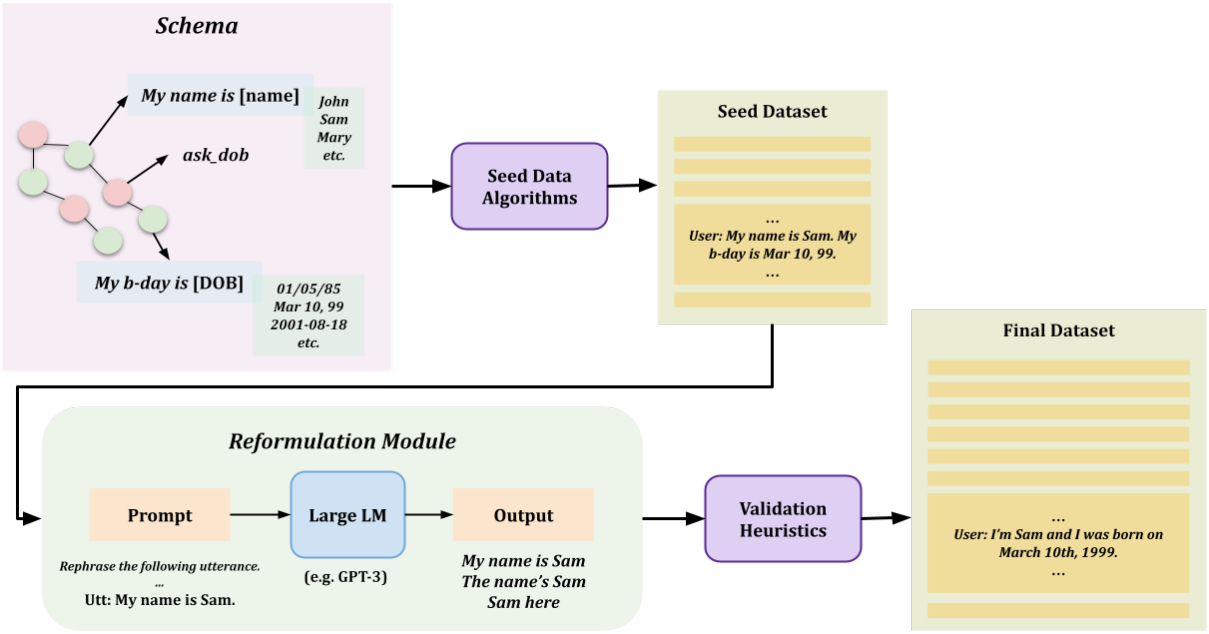


Figure 2: Visualization of LAD. (1) Domain-agnostic algorithms use the schema to create a seed dataset which conveys the necessary structural constraints. (2) Large LMs reformulate individual utterances to add linguistic diversity. (3) Validation heuristics are used to ensure adherence to the schema.

order to induce linguistic diversity, (§3.3) heuristics are used to *validate* the reformulated data to ensure adherence to the schema. LAD facilitates zero-shot generalization by explicitly leveraging the strengths of large LMs (knowledge of the distributional structure of language) without sacrificing the inductive biases (motivated by structural constraints) in the downstream neural dialog models.

### 3.1 Seed Data Creation

LAD begins by creating seed synthetic data from a given schema. This is a domain-agnostic process that aims to generate synthetic data which accurately convey the necessary structural constraints.

For **intent prediction**, the schema consists of one utterance for each intent class (sampled from the original corpus) and is used as the seed dataset. For **slot filling**, the schema consists of one manually-written template utterance and multiple slot values for each slot type. To construct the seed data: (1) begin with the utterance templates from the schema (e.g., *'My first name is {first_name}'*), (2) exhaustively combine template utterances to ensure coverage of slot type combinations, and (3) fill slot values by sampling from the schema.

The relative complexity of the structural constraints for **next action prediction**, particularly the dialog policy, necessitates a more sophisticated algorithm for generating the seed data. In order to avoid over-fitting and to ensure that the structural constraints are effectively learned by the model, it is imperative that the synthetic data produced by LAD be diverse and realistic. While linguistic diversity is induced through the reformulation with GPT-3, the synthetic dialogs created for next action prediction must also exhibit diversity of *user behavior*. The dialog policy expressed by the schema deterministically defines the *system* behavior. However, users should be able to deviate from the policy, e.g. by providing information out of turn. To account for this, Algorithm 1 in the Appendix generates a dialog by traversing the dialog policy graph and randomly combining multiple template utterances (e.g., *'**System:** What is your name? **User:** My name is John. My phone number is...'*).

### 3.2 Reformulation

To ensure that downstream neural dialog models can effectively learn the structural constraints, it is imperative that the synthetic data is sufficiently diverse. The seed synthetic data is formulaic and artificial: (1) there is a single template utterance for each user action and (2) when multiple user actions are combined they are simply concatenated. As such, the goal of the reformulation step is two-fold: (1) to induce linguistic diversity and (2) to rephrase concatenations of disjoint template utterances (*'My name is Sarah. I want to plan a party. The day*

*should be Sunday'*) into a natural utterance (*'I'm Sarah and I'd like to plan a party for Sunday.'*).

To reformulate utterances in a domain-agnostic manner, LAD leverages the in-context meta-learning abilities of GPT-3 (Brown et al., 2020). Through manual experimentation in the OpenAI Playground[3], an appropriate prompt is constructed. The prompt begins with an instruction (*'Given a set of sentences, generate 5 natural utterances that convey the same meaning.'*) and includes six examples (details can be found in the Appendix).

Rather than producing a single reformulation of the input, the chosen prompt instructs GPT-3 to generate **five** utterances. Through the examples provided in the prompt, GPT-3 learns that it should produce five *diverse* reformulations. As such, linguistic diversity is induced through both the decoding algorithm and the six examples in the prompt.

#### 3.2.1 Scalability

The cost of the GPT-3 API is approximately $0.05 USD per reformulation. In order to generate a substantial amount of synthetic data without incurring significant costs, the reformulation step of LAD must be performed in a scalable manner. The seed utterances are grouped by their intents and slot keys (e.g., 'name;date;time', 'name;date', 'date;time'). A subset of utterances in each group is reformulated. These reformulated utterances are used as templates and the slot values are randomly replaced. In this manner, the cost scales with respect to the number of distinct intent/slot combinations rather than the desired size of the synthetic dataset.

### 3.3 Validation

The seed data will always adhere to the schema and therefore *accurately* convey the necessary structural constraints. However, the reformulated utterances may not be accurate. GPT-3 may modify the intended meaning of an input utterance, for example by ignoring certain slot values. To ensure that the structural constraints are accurately expressed in the final dataset, the reformulation step of LAD filters out erroneous reformulations. For slot filling and next action prediction, this is done by ensuring that all of the slot values present in the original utterance (from the seed dataset) are also present in the reformulated utterances (produced by GPT-3).

[3]https://beta.openai.com/playground

| Original Dataset | Seed | LAD | Cost (USD) |
|---|---|---|---|
| **Intent Prediction** | | | |
| HWU64 (8955) | 64 | 800 | $19 |
| CLINC150 (15000) | 150 | 1664 | $43 |
| Banking77 (8633) | 77 | 848 | $25 |
| **Slot Filling** | | | |
| Restaurant8k (8633) | 85 | 32000 | $89 |
| **Next Action Prediction** | | | |
| STAR (1200) | 24000 | 22327 | $226 |

Table 1: Statistics for the synthetic datasets created by LAD. This table lists the size of the original dataset, the seed dataset and the final synthetic dataset produced by LAD. The last column indicates the approximate cost of using GPT-3 for each of the datasets.

### 3.4 Dataset Statistics

LAD is evaluated on five different datasets. For intent prediction, Banking77 (Casanueva et al., 2020), CLINC150 (Larson et al., 2019), and HWU64 (Liu et al., 2019a) are used. For slot filling, Restaurant8k (Coope et al., 2020) is used. For next action prediction, STAR (Mosig et al., 2020) is used. Given a human-annotated corpus, a schema is created to express the necessary constraints. LAD is then leveraged to create a synthetic dataset conditioned on the schema. Table 1 describes the size and creation cost of each of the synthetic datasets.

## 4 Experiments

This paper claims that LAD can use a schema to create a sufficiently diverse and accurate synthetic dataset, which can be used to train neural dialog models and facilitate performance gains in zero-shot settings. To validate this claim, experiments are carried out on intent prediction, slot filling and next action prediction across five datasets.

For each problem, an appropriate model from prior work is identified. The chosen models (1) exhibit strong zero-shot and few-shot generalizability, and (2) are open-source. Though LAD is not guaranteed to produce *perfectly* accurate and diverse data, the inductive biases in the chosen models make them more robust to potential errors and limitations in the synthetic data.

### 4.1 Intent Prediction

CONVBERT+*Example-Driven+Observers* (CBEO) (Mehri and Eric, 2021) is used for intent prediction. CBEO learns to predict utterance

| Model (Training Data) | BANKING77 | CLINC150 | HWU64 |
|---|---|---|---|
| CBEO (ONE-SHOT) | 31.36 | 53.96 | 43.12 |
| CBEO (ONE-SHOT + LAD) | **51.17** | **68.11** | **65.50** |
| CBEO (FULL-SHOT) | 93.83 | 97.31 | 93.03 |

Table 2: Experimental results on intent prediction. We report the accuracy of training CBEO on (1) one utterance/intent (i.e., the seed data) and (2) the synthetic data produced by LAD. For reference, we also show the results reported by Mehri and Eric (2021) obtained with full human-annotated training datasets.

| Model | F-1 |
|---|---|
| **Zero-Shot Results** | |
| CONVEX (HENDERSON AND VULIĆ, 2020) | 5.2 |
| COACH+TR (LIU ET AL., 2020) | 10.7 |
| GENSF (MEHRI AND ESKENAZI, 2021A) | 19.5 |
| GENSF + LAD | **50.9** |
| **Non Zero-Shot Results** | |
| GENSF (64 UTTERANCES) | 72.2 |
| GENSF (8633 UTTERANCES) | 96.1 |

Table 3: Experimental results on the Restaurant8k corpus. We compare GENSF + LAD with zero-shot results reported by prior work. For reference, we also show the performance of models (reported by prior work) when trained in few-shot and full-shot settings.

| Model | F-1 |
|---|---|
| **Zero-Shot Results** | |
| BERT+S (MOSIG ET AL., 2020) | 28.12 |
| SAM (MEHRI AND ESKENAZI, 2021B) | 53.31 |
| SAM + LAD | **64.36** |
| **Full-Shot Results** | |
| SAM (MEHRI AND ESKENAZI, 2021B) | 70.38 |

Table 4: Experimental results on the STAR corpus. SAM + LAD is compared with zero-shot results reported by prior work. For reference, the performance of SAM when trained on the full corpus is also shown.

intents by explicitly comparing to a set of examples. Predicting intents through an explicit non-parametric comparison to examples is an inductive bias that facilitates sample-efficient learning of the structural constraints.

The experimental results shown in Table 2 demonstrate that the synthetic data produced by LAD significantly increase performance on one-shot[4] intent prediction. LAD facilitates **15%+** accuracy improvement across all three intent prediction datasets. For intent prediction, LAD does not use any heuristics during the creation of the seed data or during the validation step. As such, these improvements can be attributed to the reformulation step, which leverages the prompt-driven generation abilities of GPT-3 (Brown et al., 2020).

### 4.2 Slot Filling

For slot filling, experiments are carried out with GENSF (Mehri and Eskenazi, 2021a) which currently has SoTA results on the Restaurant8k corpus (Coope et al., 2020), in both zero-shot and full-shot settings. GENSF reformulates slot filling as response generation in order to better leverage the capabilities of DialoGPT (Zhang et al., 2019).

As shown in Table 3, GENSF + LAD achieves a **+31.4** F-1 improvement over GENSF on the test set of Restaurant8k, without observing any examples from the corpus. GENSF + LAD learns to detect slots in the restaurant domain given only the schema, which consists of (1) a single manually written utterance for each slot type and (2) a collection of up to 20 slot values for each slot type. This significant performance improvement in zero-shot generalization validates the claim of this paper for the problem of slot filling. LAD is able to create synthetic data which effectively teaches GENSF the necessary structural constraints.

However, as shown by Mehri and Eskenazi (2021a), GENSF achieves a 72.2 F-1 score by only observing 64 human-written examples. Despite the relative success of LAD in zero-shot settings, there remains significant room for improvement.

[4]This setting is characterized as one-shot since the utterances in the schema are sampled from the respective dataset.

| Model (Training Data) | COMPLETE % | ASKS ALL % | AVOIDS REDUNDANCY % |
|---|---|---|---|
| SAM (ZERO-SHOT) | 98.02 | 76.15 | 78.90 |
| SAM (FULL-SHOT) | 98.31 | 75.69 | 80.65 |
| SAM + LAD | 98.52 | **78.39** | 79.13 |

Table 5: Results of the interactive human evaluation. We compare three models: (1) SAM (ZERO-SHOT), (2) SAM (FULL-SHOT) and (3) SAM + LAD. The three columns correspond to the three post-dialog questions: (1) task completion, (2) asking all necessary information and (3) avoiding redundancy. Results in boldface are statistically significant by one-tailed t-test ($p < 0.05$).

### 4.3 Next Action Prediction

Next action prediction is particularly challenging due to the complexity of the structural constraints. In addition to the constraints of intent prediction and slot filling, next action prediction models must also learn to follow the *dialog policy*. SAM (Mehri and Eskenazi, 2021b) learns to predict the system action by attending to a graph-based representation of the dialog policy. Explicitly attending to the dialog policy is an inductive bias that facilitates zero-shot generalization to unseen tasks.

Table 4 shows the results for three models. BERT+S (Mosig et al., 2020) trains a BERT model to attend to a rudimentary graph-based representation of the dialog policy. SAM (Mehri and Eskenazi, 2021b) improves the model architecture and introduces more expressive policy graphs. These two models are trained on the STAR corpus, which includes 24 different tasks and 24 different policy graphs. The zero-shot results are obtained by training on $n - 1$ tasks (i.e., 23) and evaluating on the remaining task, repeated 24 times. In contrast, SAM + LAD observes **no human-written dialogs** whatsoever. Instead, SAM+LAD is trained only on the synthetic dialogs produced by LAD.

In the zero-shot setting, SAM + LAD achieves an **+11.05** F-1 improvement over SAM. Furthermore, this result is only **6.02** points below the full-shot results of SAM. This significant gain further validates the claim of this paper. SAM + LAD learns the necessary structural constraints using only the synthetic data produced by LAD.

### 4.4 Interactive Human Evaluation

SAM + LAD achieves strong zero-shot results on the STAR corpus, especially relative to the performance of SAM (FULL-SHOT). This leads us to question the performance gap between these two models. Is the full-shot model better at next action prediction, or is it just better at modelling artifacts in the STAR corpus? STAR is known to have some degree of inconsistency with the policy graphs (Mosig et al., 2020). Furthermore, static evaluation is not necessarily reflective of the performance of a model in **real settings**. Because of variable user behavior, there may be a distribution shift between the STAR corpus and interactive settings. To this end, we perform an interactive human evaluation using Amazon Mechanical Turk (AMT).

Three models are evaluated: (1) SAM (ZERO-SHOT), (2) SAM (FULL-SHOT) and (3) SAM + LAD. Ten scenarios are defined, each of which consists of an objective (e.g, *'You want to plan a party'*) and slot values (e.g., `Name`: Kevin, `Date`: Sunday, `Num Guests`: 85). An AMT worker is instructed to interact with a dialog system according to the provided scenario. Upon completion of the dialog, three questions are answered:

1. Did the system successfully complete the dialog?
2. Did the system ask for all of the necessary information?
3. Did the system ask for information that you had already provided it?

The instructions (see Appendix) tell the worker to interact *naturally* (e.g., by providing information out of turn). Detailed instructions, including examples and counter-examples, are provided for the three post-dialog questions. Pre-screening is performed to ensure that AMT workers read and understood these instructions. During pre-screening, the worker must answer the post-dialog questions given two completed dialogs and the corresponding scenarios. Workers with a score of at least 5/6 qualify to participate in the interactive evaluation (45% of workers pass the pre-screening). Pre-screening is paid $0.75USD, regardless of the result. Each HIT (Human Intelligence Task) of the interactive evaluation includes five scenarios and pays $3.25USD (approx. 10 minutes). A post-hoc quality check is performed to remove erroneous

annotations. Simple heuristics are constructed to predict the post-dialog answers and any discrepancies with the annotations are manually verified. If an error is identified through manual validation, the annotation is removed. This form of validation is a necessary alternative to outlier detection or measures of inter-annotator agreement, since interactive dialogs are independent thereby making standard measures of data quality unsuitable.

1628 dialogs were collected, with at least 500 for each system. The results, shown in Table 5, demonstrate that the performance of all three models is fairly similar in interactive settings. For the second post-dialog question, SAM+LAD asks for all of the necessary slots **+2.7%** more often. Assuming that the number of observations is equal to the total number of turns, this result is statistically significant ($p < 0.05$) by one-tailed t-test.

Both SAM (FULL-SHOT) and SAM (ZERO-SHOT) are trained on human dialogs, though the latter does not observe data from the target task. In contrast, SAM + LAD is trained only on synthetic data produced by LAD. The comparable performance of SAM (ZERO-SHOT) and SAM (FULL-SHOT) is noteworthy and can potentially be explained by two facts: (1) the interactive dialogs are sampled from a different distribution (e.g., more informal, typos, more slots per utterance) from the STAR corpus, making the evaluation equally difficult for both systems, (2) SAM (ZERO-SHOT) has observed dialogs from the domain (e.g., seen `bank-balance` when evaluating on `bank-fraud-report`). Despite not observing any human dialogs, in interactive settings SAM + LAD attains zero-shot performance comparable to training on human dialogs from the STAR corpus. Though there remains significant room for improvement, the results of this interactive human evaluation demonstrate the efficacy of LAD. By leveraging the strengths of large language models to induce linguistic diversity, LAD produces synthetic data that effectively conveys the necessary structural constraints and facilitates zero-shot generalization, even in challenging interactive settings.

## 5 Related Work

### 5.1 User Simulators for Task-Oriented Dialog

The use of synthetic data in task-oriented dialog is a long-standing approach. Early dialog research leveraged user simulators for evaluation and optimization (Eckert et al., 1997; Scheffler and Young, 2000; Schatzmann et al., 2006). Schatzmann et al. (2007) propose a probabilistic agenda-based user simulator for bootstrapping a POMDB dialog system, demonstrating reasonable task completion rates. Georgila et al. (2006) train an n-gram user simulator which models both ASR and understanding errors. González et al. (2010) explicitly model user cooperativeness in a statistical user simulator.

Li et al. (2016) propose an agenda-based user simulator for training dialog policies with RL. Crook and Marin (2017) train a sequence-to-sequence model for user simulation. Kreyssig et al. (2018) introduce the neural user simulator (NUS), which trains a sequence-to-sequence network conditioned on user goals and the dialog history, outperforming existing methods on an interactive evaluation. Shi et al. (2019) carry out a comprehensive analysis of six different user simulators, with different dialog planning and generation methods. A key takeaway of this analysis is using agenda-based simulators to train RL systems generally results in higher performance on human evaluation. Lin et al. (2021) propose a domain-indepdendent transformer-based user simulator (TUS). The feature representations of TUS are domain-independent, thereby facilitating learning of cross-domain user behavior. TUS is trained on MultiWOZ (Budzianowski et al., 2018) and can effectively transfer to unseen domains.

LAD can be characterized as an agenda-based simulator, wherein the schema describes the ontology and the policy. The core novelty of LAD in the context of prior work is three-fold: (1) large LMs to induce linguistic diversity, (2) *zero-shot* domain-agnostic synthetic data creation, and (3) the schema as a standardized expression of structural constraints. LAD can potentially be further improved by incorporating strategies from prior work, such as modelling cooperativeness (González et al., 2010) or ASR errors (Georgila et al., 2006).

### 5.2 Using Large Language Models

Large language models (Brown et al., 2020; Chowdhery et al., 2022) exhibit strong language understanding, generation and reasoning abilities. Prompting is the dominant paradigm for leveraging large LMs for various downstream problems (Reynolds and McDonell, 2021; Lester et al., 2021). Madotto et al. (2021) demonstrate the efficacy of few-shot prompting for both open-domain and task-oriented dialog, with a focus on response genera-

tion and conversational parsing.

Several papers have used GPT-3 to generate synthetic data (Yoo et al., 2021; Wang et al., 2021). These approaches rely on GPT-3 to generate the labels and are not suitable for task-oriented dialog. To our knowledge, LAD is the first paper to leverage large LMs to *reformulate* utterances, in order to create synthetic data for task-oriented dialog.

## 6 Conclusion

In an effort to leverage the abilities of large LMs to facilitate zero-shot generalization in task-oriented dialog, this paper introduces **LAD**. LAD creates diverse and accurate synthetic data, in order to convey the necessary setting-specific structural constraints to neural dialog models. LAD achieves significant performance gains on zero-shot intent prediction, slot filling and next action prediction across five datasets. Furthermore, LAD is shown to perform competitively in interactive human evaluation, without observing human-annotated data.

## 7 Acknowledgements

This work was funded by a Google Research Collabs grant.

## References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. Multiwoz–a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*.

Inigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. *arXiv preprint arXiv:2003.04807*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Sam Coope, Tyler Farghly, Daniela Gerz, Ivan Vulić, and Matthew Henderson. 2020. Span-convert: Few-shot span extraction for dialog with pretrained conversational representations. *arXiv preprint arXiv:2005.08866*.

Paul A Crook and Alex Marin. 2017. Sequence to sequence modeling for user simulation in dialog systems. In *INTERSPEECH*, pages 1706–1710.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Wieland Eckert, Esther Levin, and Roberto Pieraccini. 1997. User modeling for spoken dialogue system evaluation. In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pages 80–87. IEEE.

Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075*.

Kallirroi Georgila, James Henderson, and Oliver Lemon. 2006. User simulation for spoken dialogue systems: learning and evaluation. In *Interspeech*, pages 1065–1068. Citeseer.

Meritxell González, Silvia Quarteroni, Giuseppe Riccardi, and Sebastian Varges. 2010. Cooperative user models in statistical dialog simulators. In *Proceedings of the SIGDIAL 2010 Conference*, pages 217–220.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Matthew Henderson and Ivan Vulić. 2020. Convex: Data-efficient and few-shot slot labeling. *arXiv preprint arXiv:2010.11791*.

Florian Kreyssig, Inigo Casanueva, Pawel Budzianowski, and Milica Gasic. 2018. Neural user simulation for corpus-based policy optimisation for spoken dialogue systems. *arXiv preprint arXiv:1805.06966*.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Xiujun Li, Zachary C Lipton, Bhuwan Dhingra, Lihong Li, Jianfeng Gao, and Yun-Nung Chen. 2016. A user simulator for task-completion dialogues. *arXiv preprint arXiv:1612.05688*.

Hsien-chin Lin, Nurul Lubis, Songbo Hu, Carel van Niekerk, Christian Geishauser, Michael Heck, Shutong Feng, and Milica Gašić. 2021. Domain-independent user simulation with transformers for task-oriented dialogue systems. *arXiv preprint arXiv:2106.08838*.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2019a. Benchmarking natural language understanding services for building conversational agents. *arXiv preprint arXiv:1903.05566*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zihan Liu, Genta Indra Winata, Peng Xu, and Pascale Fung. 2020. Coach: A coarse-to-fine approach for cross-domain slot filling. *arXiv preprint arXiv:2004.11727*.

Andrea Madotto, Zhaojiang Lin, Genta Indra Winata, and Pascale Fung. 2021. Few-shot bot: Prompt-based learning for dialogue systems. *arXiv preprint arXiv:2110.08118*.

Shikib Mehri and Mihail Eric. 2021. Example-driven intent prediction with observers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2979–2992.

Shikib Mehri, Mihail Eric, and Dilek Hakkani-Tur. 2020. Dialoglue: A natural language understanding benchmark for task-oriented dialogue. *arXiv preprint arXiv:2009.13570*.

Shikib Mehri and Maxine Eskenazi. 2021a. Gensf: Simultaneous adaptation of generative pre-trained models and slot filling. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 489–498.

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

Tom M Mitchell. 1980. *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research ....

Johannes EM Mosig, Shikib Mehri, and Thomas Kober. 2020. Star: A schema-guided dialog dataset for transfer learning. *arXiv preprint arXiv:2010.11853*.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2020. Soloist: Few-shot task-oriented dialog with a single pre-trained auto-regressive model. *arXiv preprint arXiv:2005.05298*.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–7.

Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-based user simulation for bootstrapping a pomdp dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152.

Jost Schatzmann, Karl Weilhammer, Matt Stuttle, and Steve Young. 2006. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The knowledge engineering review*, 21(2):97–126.

Konrad Scheffler and Steve Young. 2000. Probabilistic simulation of human-machine dialogues. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, volume 2, pages II1217–II1220. IEEE.

Weiyan Shi, Kun Qian, Xuewei Wang, and Zhou Yu. 2019. How to build user simulators to train rl-based dialog systems. *arXiv preprint arXiv:1909.01388*.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. *arXiv preprint arXiv:2104.06644*.

Zirui Wang, Adams Wei Yu, Orhan Firat, and Yuan Cao. 2021. Towards zero-label language learning. *arXiv preprint arXiv:2109.09193*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyeong Park. 2021. Gpt3mix: Leveraging large-scale language models for text augmentation. *arXiv preprint arXiv:2104.08826*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.