Turning Flowchart into Dialog: Augmenting Flowchart-grounded Troubleshooting Dialogs via Synthetic Data Generation

Haolan Zhan,   Sameen Maruf,   Lizhen Qu,   Yufei Wang
Ingrid Zukerman and **Gholamreza Haffari**
Department of Data Science & AI, Monash University, Australia
{firstname.lastname}@monash.edu

Abstract

Flowchart-grounded troubleshooting dialogue (FTD) systems, which follow the instructions of a flowchart to diagnose users' problems in specific domains (e.g., vehicle, laptop), have been gaining research interest in recent years. However, collecting sufficient dialogues that are naturally grounded on flowcharts is costly, thus FTD systems are impeded by scarce training data. To mitigate the data sparsity issue, we propose a plan-based synthetic data generation (`PlanSDG`) approach that generates diverse synthetic dialog data at scale by transforming concise flowchart into dialogues. Specifically, its generative model employs a variational-base framework with a hierarchical planning strategy that includes *global* and *local* latent planning variables. Experiments on the Flo-Dial dataset show that synthetic dialogue produced by `PlanSDG` improves the performance of downstream tasks, including flowchart path retrieval and response generation, in particular on the *Out-of-Flowchart* settings. In addition, further analysis demonstrate the quality of synthetic data generated by `PlanSDG` in paths that are covered by current sample dialogues and paths that are not covered.

1 Introduction

Flowchart-grounded Troubleshooting Dialogue (*FTD*) systems (Leake et al., 2005; Boye, 2007; Williams, 2007; Paek and Pieraccini, 2008; Janarthanam and Lemon, 2008; Wei et al., 2018; Raghu et al., 2021), which communicate with users to help them diagnose problems through the guidance of a flowchart, have been gaining interest in recent years. FTD systems face additional challenges to those faced by typical task-oriented dialogue systems (Wen et al., 2017; Budzianowski et al., 2018), e.g., FTD systems must accurately follow the instructions of a flowchart, actively detect the root cause of issues, and provide users with reasonable solutions by following an action instruction along with the *path* in a flowchart (Figure 1).
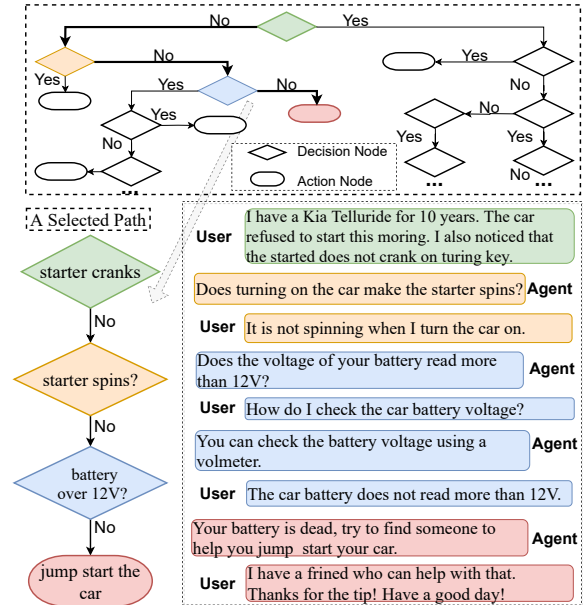

Figure 1: A sample flowchart-grounded troubleshooting dialogue. Agent follows the path of a flowchart to help user diagnose problems.

Collecting sufficiently large flowchart-related dialogue corpora for FTD is challenging, since it requires domain experts with relevant knowledge. This problem also applies to a crowd-sourced FTD corpus, such as *FloDial* (Raghu et al., 2021), whose collection still involved a great deal of human effort. Despite this, the 1,789 dialogues in *FloDial* (§ 3.1) cover only 65% of the paths in the underlying flowcharts on average (Figure 2). An alternative approach to obtaining additional dialogues could involve crawling through websites. However, most of these data obtained in this manner focus on anecdotes and subjective opinions (Dai et al., 2022), and are thus unsuitable for FTD systems.

In this paper, we propose `PlanSDG`: a **Plan**-based **S**ynthetic **D**ata **G**eneration approach that generates synthetic dialogues from flowchart paths. Specifically, `PlanSDG` takes as input a path extracted from an underlying flowchart, and generates a dialogue
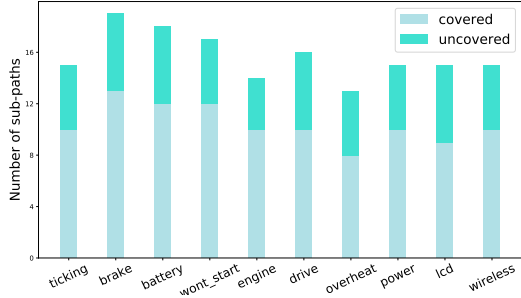
Figure 2: Statistics on the percentage (%) of (un)covered paths in the *FloDial* (containing ten flowcharts in two domains: Vehicle and Laptop) – each flowchart pertains to a specific problem. In total, more than 35% of paths are not covered by dialogue instances.

session consisting of dialogue acts and utterances. `PlanSDG` is formalised as a probabilistic generative model with structured planning latent variables, specifically *global* and *local* latent variables, that guide the generation process. The *global* latent variables are responsible for modeling the dialogue acts between the dialogue turns, providing a high-level sketch. To be able to model these global variables, we manually labeled the dialogue acts for the utterances in the *FloDial* dataset. The *local* latent variables control the diversity of generated synthetic dialogues during sentence realization.

We conducted *extrinsic* and *intrinsic* evaluations of our approach on the *FloDial* corpus, as well as follow-up ablation studies. Our extrinsic evaluation shows that the retrieval and generative models trained on the synthetic dialogues produced by `PlanSDG` achieve better performance than other augmentation methods in terms of the downstream tasks: flowchart path retrieval and response generation, particularly on the *Out-of-Flowchart* settings. Our intrinsic evaluation, which examines the quality of the synthetic dialogues, indicates that `PlanSDG` outperforms strong baseline models in term of diversity and faithfulness. Our ablation studies demonstrate the effectiveness of our proposed *global* and *local* latent planning variables. Further analysis demonstrate the quality of synthetic data generated by `PlanSDG` in *uncovered paths* that are included by flowchart but not in dialogues.

2 Plan-based Synthetic Data Generation

2.1 Task Formulation

The goal of `PlanSDG` is to take a sampled path from the flowchart, and generate a complete synthetic dialogue as well as the dialogue acts. In this paper, we only have access to a (relatively small) training set $\mathcal{T} = \{(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{y})_i\}_{i=1}^m$, where $\boldsymbol{x} = \{x_1, x_2, \ldots, x_n\}$ is a flowchart path. A path includes tuples of nodes and edges from the flowchart. Each $x_i \in \boldsymbol{x}$ on the path corresponds to a sub-dialogue $y_i = [y_{i,0}, \cdots, y_{i,|y_i|}] \in \boldsymbol{y}$, where $y_{i,j}$ is an utterance associated with a dialogue act $a_{i,j} \in \boldsymbol{a}$. For example in the flowchart path in Figure 1, the node "battery over 12V" (x_3) corresponds to the sub-dialogue starting from the turn "Does the voltage of . . ." and ending to the turn "The car battery does not . . ." ($y_{3,0}$ to $y_{3,3}$), where each turn is associated with a dialogue act.

Given a flowchart path $\boldsymbol{x}$, our proposed data augmentation method `PlanSDG` generates synthetic dialogue acts $\hat{\boldsymbol{a}}$ and dialogues turns $\hat{\boldsymbol{y}}$, and produces the synthetic dataset $\mathcal{T}_{Syn} = \{(\boldsymbol{x}, \hat{\boldsymbol{a}}, \hat{\boldsymbol{y}})_i\}_{i=1}^n$ where n could be much larger than m (e.g., 10x). Our goal is that the downstream retrieval and generative dialogue models trained using $\mathcal{T} \cup \mathcal{T}_{Syn}$ outperform the models trained using only $\mathcal{T}$.

2.2 Flowchart Path Extraction

As shown in Figure 1, the flowcharts used in this paper consist of decision nodes and action nodes. The decision nodes include a question, and they are connected with other nodes by the user responses (e.g., Yes, No). The action nodes at the bottom of the flowcharts indicate the recommended actions.

For training `PlanSDG`, we directly extract the flowchart paths for the dialogues in the training set. For syntactic data generation, to ensure full coverage for the flowchart paths, we extract the flowchart paths by *Depth-First-Search* from the top decision node to the bottom action nodes. The resulting flowchart paths are then used as the inputs for `PlanSDG`.

2.3 Synthetic Dialogue Generation

`PlanSDG` is designed to generate diverse and high-quality synthetic dialogues from the extracted flowchart paths. Even though the input flowchart paths include textual questions, user responses and final actions, conditioning only on this information could result in tedious conversations consisting of rigid sequences of question-answer pairs. Starting from a node in a flowchart, there could be many feasible open-ended dialogues. To facilitate coverage of this dialogue space, we employ intermediate latent variables in `PlanSDG`. Dialogue acts are an intuitive choice to characterise these variables, as they describe the basic function of a dialogue
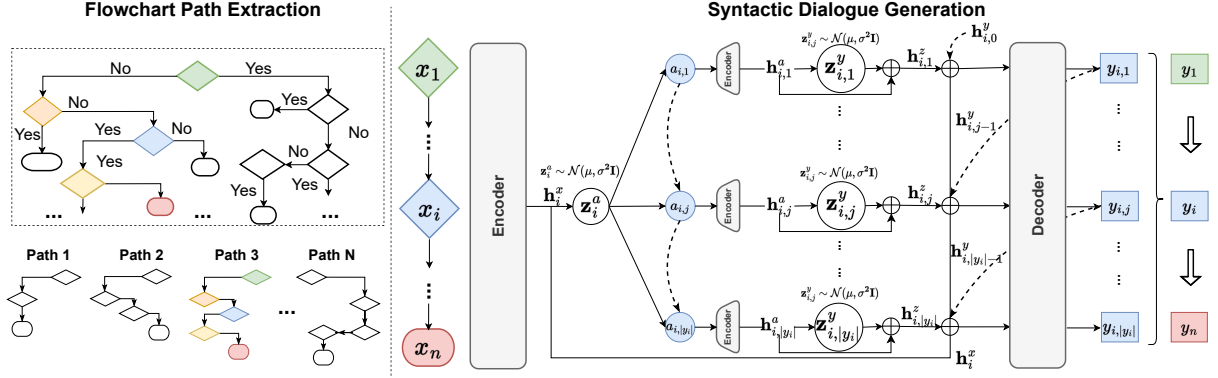
Figure 3: Detailed framework of `PlanSDG`, including path extraction and synthetic dialogue generation.

turn/utterance (e.g., inform, clarification), and reflect users' intentions (Stolcke et al., 2000; Bunt, 2011). We denote them by global latent variables z_i^a, responsible for modeling the dialogue act transition process *over the turns*. We further introduce local latent variables z_{ij}^y, responsible for generating lexically diverse utterances for each turn. As such, `PlanSDG` is formally a probabilistic generative model with structured latent variables (Figure 3), explained below in more details.

Global Planning over Dialogue Acts. We inject stochasticity into the global planning process using a continuous latent variable in each dialogue turn z_i^a, which is assumed to follow the isotropic Gaussian distribution (Kingma and Welling, 2014). We first sample z_i^a from its prior distribution $p_\theta^{z^a}(z_i^a|x_i)$, and then generate a sequence of dialogue acts auto-repressively:

$$z_i^a \sim p_\theta^{z^a}(z_i^a|x_i) \qquad (1)$$
$$a_{i,j} = p_\theta^a(.|a_{i,j-1}, x_i, z_i^a) \qquad (2)$$

where $p_\theta^a(a_{i,j}|a_{i,j-1}, z_i^a, h_i^x)$ is a 2-layer MLP with the softmax on top. We train $p_\theta^{z^a}(z_i^a|x_i)$ to approximate the posterior distribution $q_\phi(z_i^a|x_i, y_i)$ using Gaussians in the training phase. The parameters in the prior and posterior distributions, μ_a^p, σ_a^p, μ_a^q and σ_a^q, are parameterised as follows:

$$\mu_a^p = MLP_\theta^p(h_i^x),$$
$$\sigma_a^p = softplus(MLP_\theta^p(h_i^x)),$$
$$\mu_a^q = MLP_\phi^q([h_i^x, h_i^y]),$$
$$\sigma_a^q = softplus(MLP_\phi^q([h_i^x, h_i^y])),$$

where $\text{MLP}(\cdot)$ denotes a multi-layer perceptron, $softplus(\cdot)$ is a smooth approximation to ReLU, which ensures positiveness. $h_i^x = AvgPool(Enc(x_i))$ and

$h_i^y = AvgPool(Enc([y_{i,0}, \cdots, y_{i,k}]))$, which allows z_i^a to capture the global utterance information associated with x_i. Finally, the Evidence Lower Bound (ELBO) is computed as follows:

$$\mathcal{L}_{\text{global}} = -D_{KL}(q_\phi(z_i^a|x_i, y_i)||p_\theta^{z^a}(z_i^a|x_i))$$
$$+\mathbb{E}_{z_i^a \sim q_\phi}[\sum_j \log p_\theta^a(a_{i,j}|a_{i,j-1}, z_i^a, x_i)],$$

where $D_{KL}(\cdot|\cdot)$ denotes the Kullback-Leibler divergence (Kullback and Leibler, 1951).

Local Planning for Utterance Generation. Given the dialogue act $a_{i,j}$ generated from z_i^a, we focus on generating lexically diverse dialogue utterances that are faithful to the flowchart. We sample $z_{i,j}^y$ from its prior distribution conditioned on $a_{i,j}$ and x_i, as follows:

$$z_{i,j}^y \sim p_\theta^{z^y}(z_{i,j}^y|x_i, a_{i,j}) \qquad (3)$$

We train $p_\theta^{z^y}(z_{i,j}^y|x_i, a_{i,j})$ to approximate the posterior distribution $q_\phi(z_{i,j}^y|x_i, a_{i,j}, y_{i,j})$, assuming that both distributions are Gaussian. They are parameterised as follows:

$$\mu_y^p = \text{MLP}_\theta^p(h_i^x, h_{i,j}^a),$$
$$\sigma_y^p = softplus(\text{MLP}_\theta^p(h_i^x, h_{i,j}^a)$$
$$\mu_y^q = \text{MLP}_\phi^q(h_i^x, h_{i,j}^a, h_{i,j}^y),$$
$$\sigma_y^q = softplus(\text{MLP}_\phi^q(h_i^x, h_{i,j}^a, h_{i,j}^y)),$$

where $h_{i,j}^a = AvgPool(Enc(a_{i,j}))$. In contrast with global planning, here we use the ground-truth utterance $y_{i,j}$ for training to allow `PlanSDG` to focus on the local information. Finally, the ELBO for the local planning variable is:

$$\mathcal{L}_{\text{local}} =$$
$$-D_{KL}(q_\phi(z_{i,j}^y|x_i, a_{i,j}, y_{i,j})||p_\theta^{z^y}(z_{i,j}^y|x_i, a_{i,j}))$$
$$+\mathbb{E}_{z_{i,j}^y \sim q_\phi}[\log p_\theta(y_{i,j}|y_{i,j-1}, x_i, a_{i,j}, z_{i,j}^y)].$$

PlanSDG generates each utterance $y_{i,j}$ based on $\boldsymbol{h}_{i,j}^z$, x_i and $y_{i,j-1}$, as follows:

$$y_{i,j} = Dec(\boldsymbol{h}_{i,j-1}^y, \boldsymbol{h}_i^x, \boldsymbol{h}_{i,j}^z),$$

where $\boldsymbol{h}_{i,k}^z = \text{Concat}([\boldsymbol{h}_{i,j}^a, \boldsymbol{z}_{i,j}^y])$ is the concatenation of the global and local planning variables. Enc and Dec are based on the Transformer architecture, and their parameters are initialized from a pre-trained Seq2Seq model (e.g., BART).

2.4 Training Objective

To summarise, the probabilistic generative model of **PlanSDG** performs the following steps to produce a dialogue from a flowchart path $\boldsymbol{x}$. For each $x_i \in \boldsymbol{x}$ on the path, it starts by sampling the global latent variable $\boldsymbol{z}_i^a \sim p_\theta^{\boldsymbol{z}^a}(.|x_i)$, and then iteratively samples the turns $y_{i,j}$ as follows:

- Sample the dialogue act:
  $a_{i,j} \sim p_\theta^a(.|a_{i,j-1}, x_i, \boldsymbol{z}_i^a)$

- Sample the local latent variable:
  $\boldsymbol{z}_{i,j}^y \sim p_\theta^{\boldsymbol{z}^y}(.|x_i, a_{i,j})$

- Sample the utterance:
  $y_{i,j} \sim p_\theta^y(.|y_{i,j-1}, x_i, a_{i,j}, \boldsymbol{z}_{i,j}^y)$

Hence, the probability of generating a conversation and the corresponding dialogue acts given the flowchart path can be written as follows:

$$p_\theta(\boldsymbol{y}, \boldsymbol{a}|\boldsymbol{x}) = \prod_i \int d(\boldsymbol{z}_i^a) p_\theta^{\boldsymbol{z}^a}(\boldsymbol{z}_i^a|x_i) \qquad (4)$$

$$\times \prod_j \int d(\boldsymbol{z}_{i,j}^y) p_\theta^a(a_{i,j}|a_{i,j-1}, x_i, \boldsymbol{z}_i^a)$$

$$\times p_\theta^{\boldsymbol{z}^y}(\boldsymbol{z}_{i,j}^y|x_i, a_{i,j}) p_\theta^y(y_{i,j}|y_{i,j-1}, x_i, a_{i,j}, \boldsymbol{z}_{i,j}^y)$$

The overall training objective of **PlanSDG** is the sum of the ELBOs: $\mathcal{L} = \mathcal{L}_{\text{global}} + \mathcal{L}_{\text{local}}$. This is based on the variational approach to overcome the challenges of integration over the latent variables in the likelihood objective (Equation 4). We use the re-parametrization trick in (Kingma and Welling, 2014) to optimise the training objective.

3 Experiments

3.1 Setup

Dataset We use the *FloDial* dataset (Raghu et al., 2021) for our experiments. *FloDial* is a troubleshooting dialogue corpus containing 1,789 dialogues grounded on ten individual flowcharts[1]

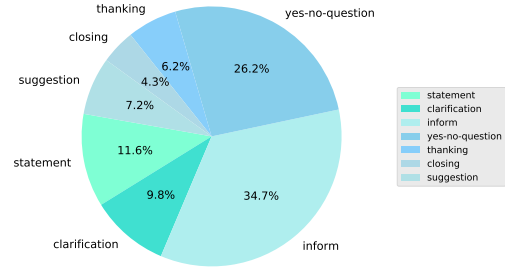[1]There is no path interaction or overlap between two individual flowcharts.

Figure 4: Statistics of dialogue act proportions in the *FloDial* dataset.

from two main domains: vehicle and laptop (five flowcharts in each domain). *FloDial* has two different settings: *In-Flowchart* and *Out-of-Flowchart*. In the *In-Flowchart* setting, both the training and test data are grounded on the same sets of flowcharts, while in the *Out-of-Flowchart* setting, the test dialogues are based on the flowcharts that are not included in the training stage.

Dialogue Act Labeling As the original *FloDial* dataset does not contain dialogue act labels, we manually label the dialogue act for each utterance. We investigated several widely-used dialogue act datasets, including Switchboard[2], AMI[3] and MultiWoz.[4] From these datasets, we select the most commonly used set of dialogue acts (i.e., cover 74.38% of the dialogue acts in these datasets) that are compatible with the *FloDial* dataset, including {*statement*, *inform*, *yes-no-question*, *clarification*, *thanking*, *closing*, *suggestion*}, and conduct annotation[5] for the *FloDial* dataset. Figure 4 shows the detailed statistics of the labeled dialogue acts.

Evaluation Settings In this paper, we conduct following evaluation: **1) Extrinsic Evaluation**: We aim to verify whether the synthetic data generated from the baselines and **PlanSDG** are useful for improving the performance of FTD. To precisely measure FTD performance, we use the same evaluation metrics as Raghu et al. (2021): Perplexity (PPL) and BLEU (Papineni et al., 2002) for response generation, and R@1 and R@5 for flowchart node retrieval.[6] **2) Intrinsic Evaluation**: We aim to confirm if our proposed model **PlanSDG** generate more diverse and faithful pseudo-dialogues than

---

[2]https://catalog.ldc.upenn.edu/LDC97S62
[3]https://groups.inf.ed.ac.uk/ami/corpus/
[4]https://github.com/budzianowski/multiwoz
[5]https://github.com/zhanhl316/flowchart-dialogue-with-DA
[6]In order to diagnose problems, at each step, the agent must retrieve the most relevant node from flowchart database.

Augmentation Model	In-Flowchart				Out-of-Flowchart			
	PPL ↓	BLEU ↑	R@1 ↑	R@5 ↑	PPL ↓	BLEU ↑	R@1 ↑	R@5 ↑
FloNet	4.93	19.36	0.834	0.957	17.08	9.53	0.529	0.765
EDA	5.67	19.65	0.837	0.956	16.84	9.79	0.535	0.772
Back-Tran	4.88	19.93	0.839	0.952	19.26	10.67	0.538	0.781
GPT-2	4.37	20.69	0.844	0.958	15.93	13.70	0.574	0.813
BART	4.52	21.11	0.852	0.965	12.48	13.94	0.581	0.826
PlanSDG w/o $\mathcal{L}_{\text{global}}$	4.61	20.75	0.847	0.963	14.25	14.17	0.583	0.829
PlanSDG w/o $\mathcal{L}_{\text{local}}$	4.48	21.06	0.843	0.956	12.45	13.83	0.579	0.832
PlanSDG	4.35*	21.18*	0.853*	0.968*	12.64	14.73**	0.609**	0.841**
DialoGPT	4.19	20.93	0.849	0.961	14.66	12.63	0.557	0.793
BlenderBot	4.06	21.26	0.847	0.960	13.06	12.89	0.562	0.804

Table 1: Extrinsic evaluation: Performance of augmented synthetic dialogue data generated by different models in *In-Domain* and *Out-of-Domain* settings. Results are based on the augmentation of **10x** the amount of data. Scores marked with "*" and "**" respectively indicate a significance of *p-value* < 0.05 and *p-value* < 0.01 in the t-test after Benjamini-Hochberg (BH) correction for false discovery rate (Benjamini and Hochberg, 1995).

the baseline models. To investigate the quality of generated synthetic data from **PlanSDG** and other baseline models, we use ROUGE (Lin, 2004) to assess fluency, Distinct (Li et al., 2016) and Self-BLEU (Zhu et al., 2018) for diversity, and Embedding Metrics (Average, Extrema, Greedy) and BART-Score (Yuan et al., 2021) for faithfulness.

Baselines Our baseline is **FloNet** (Raghu et al., 2021) which only uses the original training data $\mathcal{T}$. Given the newly generated synthetic data $\mathcal{T}_{Syn}$ from **PlanSDG** and other synthetic data generation models, we train the same **FloNet** model with $\mathcal{T} \cup \mathcal{T}_{Syn}$ under the same set of hyper-parameters. We compare **PlanSDG** with the following synthetic data generation models:

- **EDA** (Wei and Zou, 2019) is a rule-based approach by synonym replacement, random insertion, random swap, and random deletion.

- **Back-Tran** (Sennrich et al., 2016) is the classical back translation algorithm rooted from the machine translation task.

- Generic pre-trained language models including **GPT-2** (Radford et al., 2019), **BART** (Lewis et al., 2020).

- Conversational pre-trained models including **DialoGPT** (Zhang et al., 2020b) and **Blender-Bot** (Roller et al., 2021).

We use the large version for all pre-trained models. To make a fair comparison, we incorporate annotated dialogue acts for both **PlanSDG** and other synthetic data pre-trained models.

Implementation Details We utilize the state-of-the-art pre-trained text generation model BART to initialize the encoder and decoder of **PlanSDG**, for both prior and posterior, encoder and generator. For fair comparison with baseline models, we use the $\text{BART}_{\text{large}}$ for our model. In preliminary experiments, we find that fine-tuning outperforms prompt-tuning (Li and Liang, 2021) for generating valid dialogue data. For training process, we use AdamW (Loshchilov and Hutter, 2019) for gradient optimization, learning rate 0.001. batch size 8 in our experiments. We fine-tune **PlanSDG** for 50 epochs and the maximum length for utterances is set to 64. To mitigate the posterior collapse issue, we adopt the KL thresholding strategy (Kingma et al., 2016) that maximizes the KL term with a constant $\beta = 0.1$[7].

3.2 Extrinsic Evaluation

Main Results Table 1 summarizes the augmentation experiment results using 10 times (10x) for both baseline data augmentation models and **PlanSDG**. In both settings, the performance of response generation and flowchart node retrieval tasks trained with the synthetic data from **PlanSDG** are boosted up, especially in the *Out-of-Flowchart* setting. Specifically, **PlanSDG** outperforms rule-based **EDA** and naive **Back-Tran** methods by a large margin, demonstrating that widely-used data augmentation methods cannot handle the FTD situations. While comparing with strong pre-trained models (e.g, GPT-2, BART), synthetic data generated by our model have better augmentation per-

---

[7]The code will be made available upon publications.

Data Size	In-Flowchart				Out-of-Flowchart			
	PPL ↓	BLEU ↑	R@1 ↑	R@5 ↑	PPL ↓	BLEU ↑	R@1 ↑	R@5 ↑
FloNet (1x)	4.93	19.36	0.834	0.957	17.08	9.53	0.529	0.765
2x Data	5.26	20.72	0.843	0.956	13.27**	11.75*	0.546**	0.819**
5x Data	4.28	21.06*	0.851*	0.961	15.63**	14.01**	0.595**	0.837**
10x Data	4.35*	21.18*	0.853*	0.968	12.64**	14.73**	0.609**	0.841**

Table 2: Extrinsic performance. FloNet (1x) is the dataset of the baseline model (Raghu et al., 2021). 2x, 5x and 10x means that we extend the original *FloDial* training set with different amounts of synthetic data. Scores marked with "*" and "**" indicate a significance of $p < 0.05$ and $p < 0.01$ in the t-test with BH correction respectively.

Model	Uncovered path within flowchart			
	PPL ↓	BLEU ↑	R@1 ↑	R@5 ↑
FloNet	12.94	11.05	0.597	0.815
EDA	12.36	11.69	0.598	0.804
Back-Tran	13.67	12.18	0.608	0.827
GPT-2	9.82	14.61	0.632	0.854
BART	8.46	15.29	0.637	0.852
PlanSDG	8.26*	15.90**	0.654**	0.868*

Table 3: Augmentation performance on *Uncovered path* in the flowchart (*In-Flowchart* using 10x augmented synthetic data.). Scores marked with "*" and "**" indicate a significance of $p < 0.05$ and $p < 0.01$ in the t-test with BH correction respectively.

formance. We see that **PlanSDG** is more effective in the *Out-of-Flowchart* setting, though it is on-par or better than the baselines in the *In-Flowchart* setting. In the *out-of-Flowchart* setting, **PlanSDG** achieves at least 5.6% and 4.8% for BLEU and R@1 metric than baseline models. Surprisingly, model performance supported by **PlanSDG** even surpass those models supported by DialoGPT and BlenderBot which use large-scaled dialogue data for pre-training. This result suggests that with small training data, **PlanSDG** can generalize well to the domains not encountered (i.e., dialogue) in its pre-training stage.

Analysis on Synthetic Data Size Table 2 presents the augmentation performance using different size of synthetic data. FloNet (1x) only uses original training data. As shown in Table 2, the FloNet model performance keeps improving along with the data size expansion. Especially in the *Out-of-Flowchart* setting, augmentation performance improve significantly comparing to the FloNet (1x) model. These results demonstrate that **PlanSDG** can effectively learn from existing training data and produce diverse and relevant synthetic data rather than introducing noise information.

Analysis on Uncovered Path To verify the effectiveness of **PlanSDG** on uncovered path, we conduct additional experiments on a novel uncovered path setting. As discussed above, the existing training data only cover 65% of the flowchart path in the *FloDial* dataset. We split these training datasets into training (80%), as covered path, and testing (20%), as uncovered path. Table 3 summarizes the results on the uncovered path setting. **PlanSDG** achieves the best augmentation performance comparing to other augmentation baseline models. The positive results demonstrate that **PlanSDG** is capable enhance the model performance on those uncovered flowchart paths.

Ablation on Latent Variables We conduct ablation study for the components of *local* and *global* planning variables described in Section 2.3. As shown in Table 1, the elimination of *local* and *global* planning variables undermine the performance of **PlanSDG**, showing the positive contribution of these two latent variables in generating diversity and relevant synthetic data. Specifically, the ablation of *local* planning variable leads to more performance degradation than the ablation of *global* in terms of flowchart node retrieval task, showing the importance of *local* variable in controlling the diversity on sentence realization, which further impact the training on downstream tasks.

3.3 Intrinsic Evaluation

In this section, we directly verify the quality of synthetic data by using various of automatic metrics.

Automatic Metrics We show the automatic intrinsic evaluation results on synthetic dialogue in Table 4. **PlanSDG** outperforms the baselines in terms of ROUGE-L, Dist-2/3, Embedding and BART-Score. For BLEU-4 the results of **PlanSDG** are close to the baseline models. The significant improvement obtained by **PlanSDG** for Dist-2/3 indicates that our model is able to generate more

Model	BU-4 ↑	RG-L ↑	Dist-2 ↑	Dist-3 ↑	Self-B ↓	BART-S ↓	Emb (Avg/Extr/Gre) ↑
GPT-2	26.8	43.1	0.267	0.425	0.328	-2.590	88.1/68.7/84.1
BART	**29.7**	47.2	0.351	0.541	0.271	-2.164	87.2/67.5/83.3
DialoGPT	24.7	40.1	0.366	0.563	0.257	-2.328	**89.3**/62.5/82.6
BlenderBot	19.3	35.6	0.308	0.497	0.283	-2.051	82.6/59.3/78.6
w/o $\mathcal{L}_{\text{global}}$	27.3	49.1	0.382	0.574	0.249	-2.156	87.1/68.3/84.7
w/o $\mathcal{L}_{\text{local}}$	27.8	47.6	0.365	0.568	0.261	-2.321	85.7/68.2/83.8
PlanSDG	28.5	**51.2****	**0.397****	**0.602****	**0.225****	**-2.037***	86.1/**69.4***/**85.7****

Table 4: Intrinsic evaluation results for pseudo dialogue generation. The metrics BLEU-4, ROUGE-L, Distinct-2/3, Self-BLEU, BART-score and Embedding are abbreviated as BU-4, RG-L, Dist-2/3, Self-B, BART-S and Emb respectively. The best results are highlighted with **bold**. Scores marked with "*" and "**" indicate a significance of $p < 0.05$ and $p < 0.01$ in the t-test with BH correction respectively.

diverse texts than the baselines – a result of our latent variable modeling. The high scores of Embedding and BART-Score indicate that our model also has the capacity to generate utterances that are semantically coherent with the input flowchart.

Ablation on Latent Variables We first show the ablation study of different training objectives in Table 4. We observe a certain performance drop when removing global planning latent variable $\mathcal{L}_{\text{global}}$ or local planning latent variable $\mathcal{L}_{\text{local}}$ during fine-tuning. Specifically, the removal of $\mathcal{L}_{\text{local}}$ results in a significant drop in Dist-2/3 metric, showing that the local planning latent variable, together with dialogue act, is responsible for utterance diversity. We then highlight that the significance of dialogue act plays an important role in high-level sketch. The absence of $\mathcal{L}_{\text{global}}$ also results in a drop of performance in terms of BLEU-4, RG-L and Dist-2/3, showing that global planning latent variable play an important role in both relevance and diversity of the generated synthetic data. Thus, the combination of $\mathcal{L}_{\text{global}}$ and $\mathcal{L}_{\text{local}}$ guarantees the quality of generated synthetic dialogues.

3.4 Case Study

In this section, we conduct a case study towards the output of PlanSDG when given covered path and uncovered path as inputs, respectively. The output examples are given in Table 5.

Covered path We first focus on the PlanSDG outputs (Gen-1 and Gen-2) for covered flowchart path. By interpolating the latent variables, PlanSDG is able to generate diverse utterances along with the corresponding dialogue act sequences. For instance, in addition to generating a series of "yes-no-question" dialogue acts during the problem diagnosis process, PlanSDG can incorporate other dialogue acts as well, such as " clarification " and

---

Covered Flowchart paths : car won't start → starter crank? Yes → Engine fires? No → Spark to plugs? No → Spark from coil? No → 12V+ at coil primary? No → Ignition system wiring, voltage regulator.

Gen-1:
user: My family and children were very happy for the week end trip in my Honda CR-V. When everything was packed and we got seated the car would not simply start. **(statement)**
agent: After the key is turned does the starter crank? **(yes-no-q)**
user: Yes, The starter cranks when the key is turned **(inform)**
agent: When you turn the key, does the engine fire right up? **(yes-no-q)**
user: No, When the key is turned the engine does not fire up. **(inform)**
agent: Are the sparks able to reach the plug? **(yes-no-q)**
user: How do I check if it is able to reach the plug? (clarification)
agent: Hold the metal clip of the spark plug wire with a screw driver after pulling it and ask your assistant to crank it over. A bright spark will be visible to you. (suggestion)
user: No, I'm not seeing any evidence of that. The sparks aren't reaching the plug. **(inform)**
agent: Did spark come out of the ignition coil? **(yes-no-q)**
user: There is no spark near the ignition coil. **(inform)**
agent: Does the voltmeter reading show 12 volt+ at the primary coil? **(yes-no-q)**
user: No, the voltmeter reading does not show 12 volt + at the primary coil. **(inform)**
agent: There are a couple of suggestions. Either your car ignition system wiring or the voltage regulator is at fault. **(suggestion)**

Gen-2:
user: I am unable to start my Lincoln Continental. I think the problem is major, because I've only had the car for a week. **(statement)**
agent: Does turning the key make the starter crank? **(yes-no-q)**
user: It cranks when I turn the key. **(inform)**
agent: Can you tell me if the engine fires on turning the key? **(yes-no-q)**
user: The engine doesn't fire on turning the key. **(inform)**
agent: Is the spark reaching the plugs? **(yes-no-q)**
user: Spark isn't reaching the plugs. **(inform)**
agent: Is there any kind of spark from the ignition coil? **(yes-no-q)**
user: It's not coming from there. **(inform)**
agent: Is the reading more than 12 volt at the primary coil? **(yes-no-q)**
user: No, the reading is not more than 12 volt at the primary coil. **(inform)**
agent: The problem is the car ignition system wiring or voltage regulator is not working. Hope I was helpful. **(suggestion)**

Uncovered Flowchart paths : laptop wireless problem. → See Wireless Network? No → Switch on, enabled? Yes → outer active, in range? No → Get router wireless activity, LED blinking, try laptop in same room
user: I am unable to connect to the wireless network with my Lenovo. Any solution I can fix this? **(statement)**
agent: Can you see the wireless network you want to connect to? **(yes-no-q)**
user: How do I check if my laptop to see the wireless network? **(clarification)**
agent: Click on the network icon, then you can see it. **(Suggestion)**
user: Sorry, man, it isn't showing up. **(inform)**
agent: Ok, Is the wireless adapter switched ON? **(yes-no-q)**
user: Yes, of course. It's switched on. **(Inform)**
agent: Is the router in range and active? **(yes-no-q)**
user: No, I checked my network router and it's within range and working. **(inform)**
agent: Make sure the router is turned ON and its wireless activity LED is blinking. You should be able to connect in this way. **(suggestion)**

Table 5: Pseudo dialogue generation case by our PlanSDG model, including covered path and uncovered path.

" suggestion ". This will further lead to the diversity in their corresponding utterance generation. In addition, although being diverse, both example dialogues still maintain high faithfulness towards the input flowchart path.

Model	Faith.	Rel.	Info.	*kappa*
EDA	1.37	1.85	2.09	0.64
Back-Tran	1.62	2.27	2.18	0.59
GPT-2	2.24	2.53	**2.65**	0.56
BART	2.19	2.59	2.16	0.59
`PlanSDG`	**2.33**	**2.60**	2.54	0.57

Table 6: Human Evaluation. Annotators are required to judge each instances individually generated by baselines and our model.

Uncovered Paths As only 65% flowchart paths are covered in the *FloDial* training data, we conduct a further qualitative analysis to explore whether `PlanSDG` can generate acceptable synthetic dialogues for those *uncovered* paths. As shown in the bottom case in Table 5, we can tell from the example that basic requirements such as fluency, naturalness, and faithfulness have been fulfilled. We hypothesise that, through fine-tuning on those covered dialogue instances, dialogue systems trained on `PlanSDG` augmented data acquire and memorize relevant domain knowledge in flowcharts. Therefore, these dialogue systems will likely to have better performance compared to the ones which have not seen training data instances for the uncovered flowchart paths.

3.5 Human Evaluation

We have shown that our proposed `PlanSDG` method can achieve better performance in both extrinsic and intrinsic evaluations. However, the automatic metrics do not necessarily reflect human preference of the generated text. We therefore select 150 output samples for each baseline synthetic models and `PlanSDG` model. For each individual sample, we ask three annotators to judge from three aspects: *Faithfulness*, *Relevance* and *Informativeness*. The scale ranges from 0 (low) to 3 (high). Table 6 summarizes human evaluation results. The kappa scores indicate that the annotators came to a fair agreement in the judgement. Compared to baseline models, our `PlanSDG` approach achieves higher performance on its generated synthetic dialogues. Thus, synthetic data from `PlanSDG` also aligns well with human preferences.

4 Related Work

4.1 Troubleshooting Dialogue Systems

Troubleshooting dialogues typically appear in problem-solving scenarios between a novice and an expert (Boye, 2007; Williams, 2007; Janarthanam and Lemon, 2008). In such scenarios, experts with domain knowledge help novices by asking a series of questions to identify the problem, while the novice mostly supplies answers. Recently, Wei et al. (2018) built an end-to-end system for patient diagnosis, and a flowchart-grounded troubleshooting dialogue scenario was proposed by (Raghu et al., 2021). However, these methods are only explored in limited domains and datasets (e.g., computer, car), while `PlanSDG` is a general approach to synthesize pseudo dialogues.

4.2 Data Augmentation for Dialogue

Data augmentation for dialogue-related tasks has been explored in several previous works: Quan and Xiong (2019) presented sentence and word-level data augmentation approaches for end-to-end task-oriented dialogues; Hou et al. (2018) presented a seq2seq framework to augment dialogue utterances for dialogue language understanding, including a ranking system to produce diverse utterances; Zhang et al. (2020a) proposed a Multi-Action Data Augmentation (MADA) model, which uses dialog states to summarize the dialog history, and then maps dialog states to their system actions. Data augmentation methods for spoken dialogue and language understanding, including generative latent variable models, were investigated in (Hou et al., 2018; Kim et al., 2019; Yoo et al., 2019). However, most of the previous works focus on data augmentation for discriminative tasks. Kann et al. (2022) used retrieval-based data augmentation to improve response generation performance in open-domain dialogues, which heavily rely on relevant external resource. Given the limited relevant external resource in FTD, the retrieval-based data augmentation method cannot be applied for FTD systems.

5 Conclusions

In this paper, we explore the synthetic dialogue generation as a data augmentation approach with pre-trained model for flowchart-grounded troubleshooting dialogue systems. In further, in order to incorporate dialogue-specific features efficiently, we present a planning-based generative model `PlanSDG` for generating synthetic dialogues on troubleshooting dialogue task. The generated augmented dataset is then used to train an FTD systems. Experiments on the *FloDial* benchmark show the effectiveness of our proposed method. In the future, we plan to generalise our method to more complex dialogues, and apply it to other tasks.

Ethics Statement

We emphasize several ethical consideration in this work. First, we would like to achknowledge the efforts of crowd-workers and annotators throughout the dataset annotation and human evaluation processes. This study underwent a thorough review and received approval from an internal board. Every annotator received a compensation of 25 AUD per hour during the annotation and evaluation stages. The associated dataset is strictly for research purpose only.

References

Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300.

Johan Boye. 2007. Dialogue management for automatic troubleshooting and other problem-solving applications. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, pages 247–255, Antwerp, Belgium. Association for Computational Linguistics.

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.

Harry Bunt. 2011. The semantics of dialogue acts. In *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011)*.

Zhuyun Dai, Arun Tejasvi Chaganty, Vincent Y Zhao, Aida Amini, Qazi Mamunur Rashid, Mike Green, and Kelvin Guu. 2022. Dialog inpainting: Turning documents into dialogs. In *International Conference on Machine Learning*, pages 4558–4586. PMLR.

Yutai Hou, Yijia Liu, Wanxiang Che, and Ting Liu. 2018. Sequence-to-sequence data augmentation for dialogue language understanding. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1234–1245, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Srinivasan Janarthanam and Oliver Lemon. 2008. User simulations for online adaptation and knowledge-alignment in troubleshooting dialogue systems. *Semantics and Pragmatics of Dialogue (LONDIAL)*, page 45.

Katharina Kann, Abteen Ebrahimi, Joewie Koh, Shiran Dudy, and Alessandro Roncone. 2022. Open-domain dialogue generation: What we can do, cannot do, and should do next. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 148–165, Dublin, Ireland. Association for Computational Linguistics.

Hwa-Yeon Kim, Yoon-Hyung Roh, and Young-Kil Kim. 2019. Data augmentation by data noising for open-vocabulary slots in spoken language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 97–102, Minneapolis, Minnesota. Association for Computational Linguistics.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. 2016. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

David B Leake, Steven Bogaerts, Michael Evans, Rick McMullen, Michael Oder, and Alejandro Valerio. 2005. Using cases to support divergent roles in distributed collaboration. In *FLAIRS Conference*, pages 117–122.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. 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 4582–4597, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Tim Paek and Roberto Pieraccini. 2008. Automating spoken dialogue management design using machine learning: An industry perspective. *Speech communication*, pages 716–729.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Jun Quan and Deyi Xiong. 2019. Effective data augmentation approaches to end-to-end task-oriented dialogue. In *2019 International Conference on Asian Language Processing (IALP)*, pages 47–52. IEEE.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Dinesh Raghu, Shantanu Agarwal, Sachindra Joshi, and Mausam. 2021. End-to-end learning of flowchart grounded task-oriented dialogs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4348–4366, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–374.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Zhongyu Wei, Qianlong Liu, Baolin Peng, Huaixiao Tou, Ting Chen, Xuanjing Huang, Kam-fai Wong, and Xiangying Dai. 2018. Task-oriented dialogue system for automatic diagnosis. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–207, Melbourne, Australia. Association for Computational Linguistics.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449, Valencia, Spain. Association for Computational Linguistics.

Jason Williams. 2007. Applying POMDPs to dialog systems in the troubleshooting domain. In *Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies*, pages 1–8, Rochester, NY. Association for Computational Linguistics.

Kang Min Yoo, Youhyun Shin, and Sang-goo Lee. 2019. Data augmentation for spoken language understanding via joint variational generation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 7402–7409. AAAI Press.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.

Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020a. Task-oriented dialog systems that consider multiple appropriate responses under the same context. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9604–9611. AAAI Press.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing

Liu, and Bill Dolan. 2020b. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 1097–1100. ACM.

A Appendix

A.1 Derivation of Variational Lower Bound

$$\log p_\theta(\mathbf{a}, \mathbf{y}|\mathbf{x})$$

$$= \log \int_{\mathbf{z}_a} \int_{\mathbf{z}_y} p_\theta(\mathbf{a}|\mathbf{z}_a, \mathbf{x}) \cdot$$

$$p_\theta(\mathbf{y}|\mathbf{z}_y, \mathbf{a}, \mathbf{x}) p_\phi(\mathbf{z}_y|\mathbf{a}, \mathbf{x}) p_\phi(\mathbf{z}_a|\mathbf{x}) d_{\mathbf{z}_a}$$

$$= \log \int_{\mathbf{z}_a} p_\theta(\mathbf{a}|\mathbf{z}_a, \mathbf{x}) p_\phi(\mathbf{z}_a|\mathbf{x}) \frac{q_\phi(\mathbf{z}_a|\mathbf{x}, \mathbf{y})}{q_\phi(\mathbf{z}_a|\mathbf{x}, \mathbf{y})} \cdot$$

$$\int_{\mathbf{z}_y} p_\theta(\mathbf{y}|\mathbf{z}_y, \mathbf{a}, \mathbf{x}) p_\phi(\mathbf{z}_y|\mathbf{a}, \mathbf{x}) \frac{q_\phi(\mathbf{z}_y|\mathbf{x}, \mathbf{a}, \mathbf{y})}{q_\phi(\mathbf{z}_y|\mathbf{x}, \mathbf{a}, \mathbf{y})} d_{\mathbf{z}_x}$$

$$= \log \int_{\mathbf{z}_a} p_\theta(\mathbf{a}|\mathbf{z}_a, \mathbf{x}) p_\phi(\mathbf{z}_a|\mathbf{x}) \frac{q_\phi(\mathbf{z}_a|\mathbf{x}, \mathbf{y})}{q_\phi(\mathbf{z}_a|\mathbf{x}, \mathbf{y})} \cdot$$

$$\mathbb{E}_{q_\phi(\mathbf{z}_y|\mathbf{x}, \mathbf{a}, \mathbf{y})} \left[\frac{p_\theta(\mathbf{y}|\mathbf{z}_y, \mathbf{a}, \mathbf{x}) p_\phi(\mathbf{z}_y|\mathbf{a}, \mathbf{x})}{q_\phi(\mathbf{z}_y|\mathbf{x}, \mathbf{a}, \mathbf{y})} \right] d_{\mathbf{z}_x}$$

$$= \log \mathbb{E}_{q_\phi(\mathbf{z}_a|\mathbf{x}, \mathbf{y})} \{ \frac{p_\theta(\mathbf{a}|\mathbf{z}_a, \mathbf{x}) p_\phi(\mathbf{z}_a|\mathbf{x})}{q_\phi(\mathbf{z}_a|\mathbf{x}, \mathbf{y})} \cdot$$

$$\mathbb{E}_{q_\phi(\mathbf{z}_y|\mathbf{x}, \mathbf{a}, \mathbf{y})} \left[\frac{p_\theta(\mathbf{y}|\mathbf{z}_y, \mathbf{a}, \mathbf{x}) p_\phi(\mathbf{z}_y|\mathbf{a}, \mathbf{x})}{q_\phi(\mathbf{z}_y|\mathbf{x}, \mathbf{a}, \mathbf{y})} \right] \}$$

$$\geq \mathbb{E}_{q_\phi(\mathbf{z}_a|\mathbf{x}, \mathbf{y})} \{ \log \frac{p_\theta(\mathbf{a}|\mathbf{z}_a, \mathbf{x}) p_\phi(\mathbf{z}_a|\mathbf{x})}{q_\phi(\mathbf{z}_a|\mathbf{x}, \mathbf{y})} +$$

$$\mathbb{E}_{q_\phi(\mathbf{z}_y|\mathbf{x}, \mathbf{a}, \mathbf{y})} \left[\frac{p_\theta(\mathbf{y}|\mathbf{z}_y, \mathbf{a}, \mathbf{x}) p_\phi(\mathbf{z}_y|\mathbf{a}, \mathbf{x})}{q_\phi(\mathbf{z}_y|\mathbf{x}, \mathbf{a}, \mathbf{y})} \right] \}$$

$$\approx -KL(q_\phi(\mathbf{z_a}|\mathbf{x}, \mathbf{y})||p_\theta(\mathbf{z_a}|\mathbf{x}))$$

$$+ \mathbb{E}_{\mathbf{z_a} \sim q_\phi}[\log p_\theta(a|\mathbf{z_a}, \mathbf{x})]$$

$$- KL(q_\phi(\mathbf{z_y}|\mathbf{x}, a, \mathbf{y})||p_\theta(\mathbf{z_y}|\mathbf{x}, a))$$

$$+ \mathbb{E}_{\mathbf{z_y} \sim q_\phi}[\log p_\theta(\mathbf{y}|\mathbf{x}, a, \mathbf{z}_y)]$$

Domain	Vehicle				
	ticking	brake	battery	wont_start	engine
#Dialog	178	188	196	174	168
#path	15	19	18	17	14
Domain	Laptop				
	drive	overheating	power	lcd	wireless
#Dialog	192	186	188	178	196
#path	16	13	15	15	15

Table 7: #Dialog and #sub-path denote the number of dialogue session, and the number of sub-paths of each corresponding flowchart.

A.2 Details about *FloDial* Dataset

The *FloDial* dataset is collected for the troubleshooting situations, where the interactions between user and agent are carried to diagnose user's problem in specific domain. *FloDial* contains two main domain: vehicle and laptop. Each domain contains 5 sub-problems. For each sub-problem, there is a corresponding flowchart. Dialogues are conducted based on these flowcharts. Details about each sub-problems and flowchart are shown in Table 7. *FloDial* contains 1,789 dialogue sessions in total. In the experiments of *FloDial* paper, they construct two settings: *In-Flowchart* and *Out-of-Flowchart* settings. The test set of *In-Flowchart* setting contains the dialogue in 8 sub-problems (including ticking, brake, battery, wont_start, drive, overheating, power and lcd), which maintains the same domain with training set. Beside, the test set of *Out-of-Flowchart* setting only contains 2 sub-problems (engine, wireless), while all other 8 sub-problems are treated as training set. An example of flowchart in *car_wont_start* domain is shwon in Figure 5

Besides, as the original *FloDial* dataset does not contain any dialogue act information, we manually label the dialogue act for each dialogue utterance. The selection of dialogue acts is based on the investigation on previous work, including Switchboard (https://catalog.ldc.upenn.edu/LDC97S62), AMI (https://groups.inf.ed.ac.uk/ami/corpus/), MultiWoz (Budzianowski et al., 2018) and etc. Finally, we chose seven most frequent dialogue, which also compatible with the *FloDial* dataset. These dialogue acts include: {statement, inform, yes-no-question, clarification, thanking, closing and suggestion}. The percentage of each dialogue act in the *FloDial* is: statement: 11.6%, inform: 34.7%, yes-no-question: 26.2%, clarification: 9.8%, thanking: 6.2%, closing: 4.3% and suggestion: 7.2%.
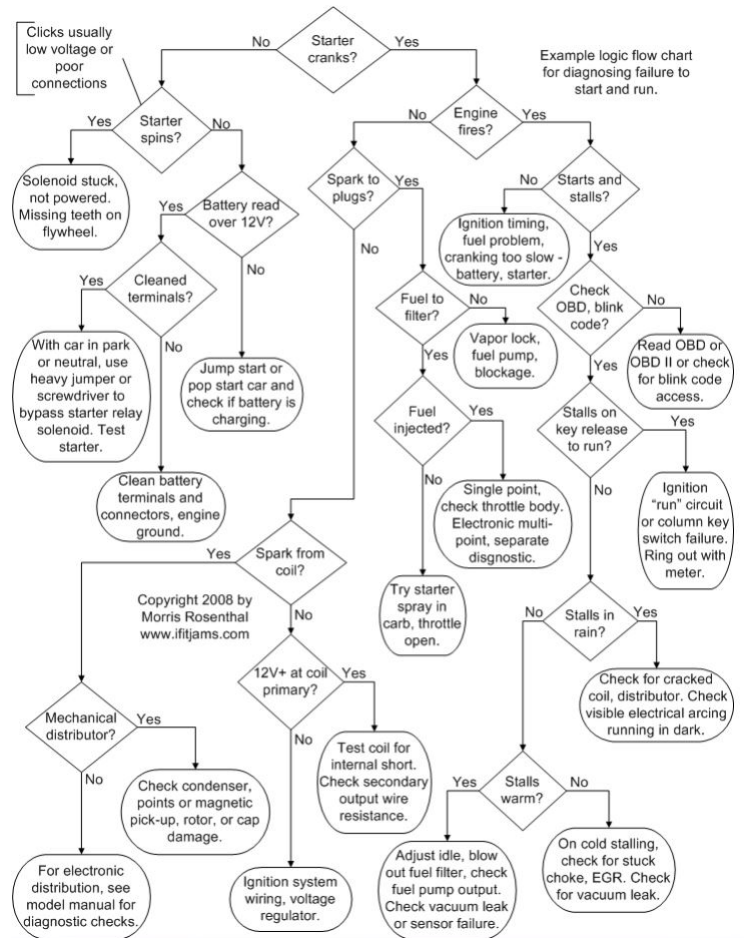

Figure 5: The flowchart example of *car_wont_start* domain. The figure is directly downloaded from the website:https://www.ifitjams.com/, the original source of *FloDial* dataset.