Self-training with Two-phase Self-augmentation for Few-shot Dialogue Generation

Wanyu Du Hanjie Chen Yangfeng Ji
Department of Computer Science
University of Virginia
Charlottesville, VA 22904
{wd5jq,hc9mx,yangfeng}@virginia.edu

Abstract

In task-oriented dialogue systems, response generation from meaning representations (MRs) often suffers from limited training examples, due to the high cost of annotating MR-to-Text pairs. Previous works on self-training leverage fine-tuned conversational models to automatically generate pseudo-labeled MR-to-Text pairs for further fine-tuning. However, some self-augmented data may be noisy or uninformative for the model to learn from. In this work, we propose a two-phase self-augmentation procedure to generate high-quality pseudo-labeled MR-to-Text pairs: the first phase selects the most informative MRs based on model's prediction uncertainty; with the selected MRs, the second phase generates accurate responses by aggregating multiple perturbed latent representations from each MR. Empirical experiments on two benchmark datasets, FEWSHOTWOZ and FEWSHOTSGD, show that our method generally outperforms existing self-training methods on both automatic and human evaluations.[1]

1 Introduction

In task-oriented dialogue systems, a natural language generation (NLG) module is an essential component: it maps structured dialogue meaning representations (MRs) into natural language responses. The NLG module has a great impact on users' experience because it directly interacts with users using text responses (Wen et al., 2015; Rastogi et al., 2020a; Kale and Rastogi, 2020; Peng et al., 2020). However, in real-world applications, developers often only have a few well-annotated data and confront a high data collection cost in specific domains. This real-world challenge makes building an NLG module in the low-data setting a valuable research problem (Kale and Rastogi, 2020; Chen et al., 2020; Peng et al., 2020).

[1]Please check the code, data, and evaluation scripts of this work at: https://github.com/wyu-du/Self-Training-Dialogue-Generation

	Self-augmented Data	$\mathbb{E}[p_\theta]$	$Var[p_\theta]$
1	request (ref = ?) & i am sorry i do not have any restaurants with those criteria	low	low
2	inform (choice = many) @ request (food = ?) & there are many restaurants that serve vegetarian food	low	high
3	inform (food = seafood) & it is seafood	high	low
4 ✓	inform (choice = several) @ request (area = ?) & there are several restaurants you'd like to dine in?	high	high

Table 1: Examples of our self-augmented data and data selection strategy. text is the input MR (e.g. *request* is the dialogue intent, and *(ref = ?)* is the slot-value pair of the current intent). The model p_θ generates synthetic dialogue response conditioning on the text. For each self-augmented data, a **low** predictive mean $\mathbb{E}[p_\theta]$ indicates that the model finds the augmented data "too noisy" (e.g. out-of-domain or invalid response), and a **low** predictive variance $Var[p_\theta]$ indicates that the model finds the augmented data "too certain" (e.g. uninformative response). In this work, we propose to select examples with **high** $\mathbb{E}[p_\theta]$ and **high** $Var[p_\theta]$.

While language models have been widely adopted to build the NLG module in task-oriented dialogue systems, they usually require thousands of MR-to-Text pairs for learning the domain-specific knowledge (Wen et al., 2016; Zhu et al., 2019; Yang et al., 2021; Lee, 2021). To collect more training data under a feasible budget, previous works propose three general approaches: (1) designing handcraft rules to augment new data, which is hard to scale up (Wei and Zou, 2019; Feng et al., 2020); (2) building task-specific data retriever to search related data, which may overfit on the few training data (Xu et al., 2021); or (3) leveraging pre-trained language models to generate new data, which may generate "too noisy" data (Peng et al., 2021; Fabbri et al., 2021; Heidari et al., 2021).

Ideally, the augmented data should help the

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 2770–2784
December 7-11, 2022 ©2022 Association for Computational Linguistics

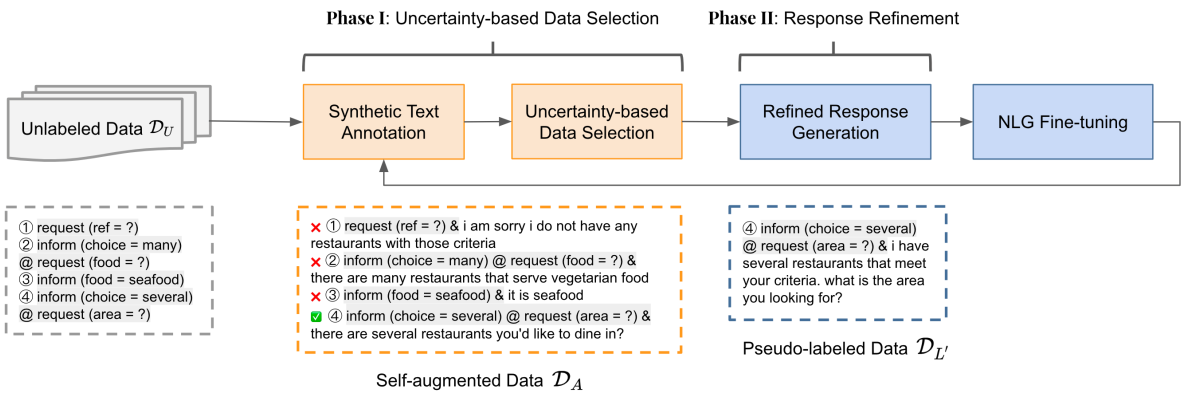

Figure 1: Our two-phase self-augmentation (SA^2) self-training framework for few-shot MR-to-Text generation.

model better learn the domain-specific knowledge. However, some augmented data can be "too noisy" that leads the model to learn irrelevant or inappropriate data patterns. This phenomenon is also described as negative transfer in other works (Chen et al., 2011; Wang et al., 2019; Meftah et al., 2021; Feng et al., 2021). To address this challenge, some works leverage human judgements to filter out the "too noisy" augmented data, which are difficult to scale up across different domains and tasks (Peris and Casacuberta, 2018; P.V.S and Meyer, 2019). Other works train task-specific discriminators to pick up the valid augmented data, which are likely to overfit in the low-data setting (Mi et al., 2021; Xu et al., 2021; Bakshi et al., 2021; Heidari et al., 2021; Mehta et al., 2022).

In this work, we propose to address the issue of selecting high-quality self-augmented examples with a two-phase procedure, where each phase will take care of selecting inputs and generating outputs independently. As illustrated in Figure 1, the first phase evaluates input MRs with model's prediction uncertainty, aiming at selecting input examples that are informative to the current model. Specifically, for each input MR, we let the current model generate a response, and then apply the Monte Carlo Dropout method (Gal and Ghahramani, 2016) to estimate the predictive mean $\mathbb{E}[p_\theta]$ and predictive variance $Var[p_\theta]$ of the generated response. In uncertainty quantification (Gal, 2016), high predictive mean indicates that the model is familiar with this input (i.e. in-domain data) and high predictive variance reflects that the model is sensitive to this input (i.e. informative data). Hence, we propose to select input MRs with high predictive mean and variance. Note that our uncertainty-based data selection strategy neither requires training additional neural models to select the valid data (Bakshi et al., 2021; Heidari et al., 2021; Mehta et al., 2022), nor need to calculate the data statistics across all training epochs and re-train the model overall again (Swayamdipta et al., 2020). The second phase aims at further improving the quality of the selected data. We adopt an idea from contrastive representation learning (Gao et al., 2021) and use the aggregation of randomly perturbed latent representations to help the model produce more accurate responses. The combination of these two phases guarantees the proposed method selects more informative MR inputs and generates less noisy responses for further model fine-tuning.

In summary, the contributions of this work are as follows:

1. Proposing a novel self-training algorithm for the few-shot MR-to-Text generation problem in task-oriented dialogue systems, which applies a two-phase self-augmentation strategy to identify informative MRs and generate accurate responses for further fine-tuning.
2. Showing that the proposed method generally outperforms other few-shot NLG baselines on two benchmark datasets, FEWSHOTWOZ (Peng et al., 2020) and FEWSHOTSGD (Xu et al., 2021) in both automatic and human evaluations.
3. Conducting in-depth empirical analysis on key components of the proposed few-shot self-training framework: the pre-trained language model, the data selection strategy, and the model training configurations.

2 Related Works

Task-oriented dialogue generation. Previous NLG methods generate system responses by: (1)

designing handcraft response templates and filling in slot-value pairs from system actions, or (2) building data-driven neural models, which encode systems actions into latent feature representations and decode natural language responses with more diversity in realization. However, both approaches cause high data collection costs. The template-based methods (Langkilde and Knight, 1998; Cheyer and Guzzoni, 2006) require collecting a comprehensive set of templates to cover all possible combinations of dialog acts and slot-value pairs, while data-driven methods (Wen et al., 2015, 2017; Zhu et al., 2019) require collecting thousands of system action and response pairs to ensure the neural model generating fluent responses.

Few-shot NLG. Recent works on few-shot NLG mainly focus on developing or adapting pre-trained language models. Peng et al. (2020) presents the first few-shot NLG benchmark for task-oriented dialog systems, and develops a pre-trained language model which can be fine-tuned with only a few domain-specific labels to adapt to new domains. Chen et al. (2020) applies the switch mechanism to combine the information from both input data and pre-trained language models, which achieves good performance in table-to-text generation tasks. Chang et al. (2021) studies the training data selection strategies in few-shot NLG, and finds that clustering-based selection strategy consistently helps generative models get better performance than randomly sampling.

Self-training for NLG. There has been some works applying the self-training technique to improve the model's generalization ability in NLG tasks. Some works (Mi et al., 2021; Xu et al., 2021) leverage the self-training framework to pseudo-label the unlabeled data and select the training data based on the confidence score from a single student model. Other works (Kedzie and McKeown, 2019; He et al., 2020) show that the noisy self-training is able to utilize unlabeled data and improve the performance of the supervised baseline. However, their observations come from large-scale training datasets, which may not necessarily hold in the few-shot data setting, because a single Transformer-based model may heavily overfit on the few-shot training data in the early iteration.

We also find some works (Bakshi et al., 2021; Heidari et al., 2021; Mehta et al., 2022) leverage generation models to produce pseudo-labeled data. However, they train additional neural models to select the pseudo-labeled data. Bakshi et al. (2021) and Heidari et al. (2021) use the reconstruction loss from a fine-tuned BART model (Lewis et al., 2020) to select the pseudo-labeled data. Besides, Mehta et al. (2022) leverage a fine-tuned BLEURT model (Sellam et al., 2020) with a selection threshold to select pseudo-responses for self-training. Intuitively, the pseudo-labeled data should bring new domain-specific knowledge to the model. While prior works select the pseudo-labeled data using an independent neural model, we propose to select the pseudo-labeled data using the generation model itself and eliminate the requirement for training additional models.

Data selection strategies. Some works in active learning leverage human judgments to select the augmented data. Peris and Casacuberta (2018); P.V.S and Meyer (2019) design data selection functions to select a subset of representative unlabeled data for humans to annotate, and get better model performance by leveraging human annotation. However, the additional requirement of human judgments will increase the difficulty of adapting the method across different domains. Another work (Swayamdipta et al., 2020) leverages the model training dynamics to categorize and select the data, but their method requires massive ground-truth labeled data. In contrast, our self-training framework does not require additional human judgments or massive ground-truth labeled data, which can be easily adapted to different tasks across different domains.

3 Proposed Method

In task-oriented dialogue systems, the NLG module translates a structured dialogue meaning representation $\mathcal{A}$ into a natural language response $\boldsymbol{x} = \{x_1, ..., x_T\}$. One structured dialogue meaning representation $\mathcal{A}$ consists of K dialogue intents and a list of slot-value pairs for each intent:

$$\mathcal{A} = \{\mathcal{I}_k, (s_{k,1}, v_{k,1}), ..., (s_{k,P_k}, v_{k,P_k})\}_{k=1}^{K} \quad (1)$$

where the dialogue intent $\mathcal{I}_k$ indicates different types of system actions and the slot-value pairs $\{(s_{k,i}, v_{k,i})\}_{i=1}^{P_k}$ shows the category names and their content information to be expressed in the response. For example, *inform (area = west; choice = many)*, where *inform* is the dialogue intent, *area* and *choice* are the slot names, *west* and *many* are the slot values.

We define $p_\theta(\boldsymbol{x} \mid \mathcal{A})$ as the generation model that generates the response $\boldsymbol{x}$ in an auto-regressive way conditioning on $\mathcal{A}$:

$$p_\theta(\boldsymbol{x} \mid \mathcal{A}) = \prod_{t=1}^{T} p_\theta(x_t \mid x_{1:t-1}, \mathcal{A}) \qquad (2)$$

where θ is the model parameter. A typical way of learning θ is by maximizing the log-likelihood of the conditional probabilities in Equation 2 over the original training set $\mathcal{D}_L$:

$$\mathcal{L}_\theta(\mathcal{D}_L) = \sum_{n=1}^{|D_L|} \sum_{t=1}^{T_n} \log p_\theta(x_{t,n} \mid x_{1:t-1,n}, \mathcal{A}_n) \qquad (3)$$

In the few-shot MR-to-Text generation setting, the size of training data $|D_L|$ is a small number (e.g. ≤ 50).

3.1 Self-training with Two-phase Self-augmentation (SA^2)

The SA^2 self-training algorithm starts from a warm-up stage, where a base generation model is trained on the original training set $\mathcal{D}_L$ for a few epochs. Then, in each iteration of self-training, the algorithm consists of four steps: synthetic text annotation, uncertainty-based data selection, response refinement, and model fine-tuning.

The synthetic text annotation uses the current model to generate synthetic text responses based on input MRs and constructs a preliminary version of self-augmented data $\mathcal{D}_A$. Next, the data selection uses the prediction uncertainty of the current model on the synthetic responses to select informative MRs in $\mathcal{D}_A$, which is the *first phase* of self-augmentation. Given the selected MRs, the *second phase* of self-augmentation is to generate more accurate text responses via aggregating multiple latent representations from model parameters with different dropout masks, which produces the pseudo-labeled data $\mathcal{D}_{L'}$. Finally, the current model is fine-tuned with both the original training set $\mathcal{D}_L$ and the pseudo-labeled dataset $\mathcal{D}_{L'}$.

The detailed procedure of SA^2 self-training algorithm is demonstrated in algorithm 1. We describe the proposed uncertainty-based data selection method in §3.2 and response refinement method in §3.3 respectively.

Algorithm 1: SA^2 Self-training Algorithm

Input: The original training set $\mathcal{D}_L$, in-domain MRs $\mathcal{D}_U$, base generation model p_θ, number of self-training iterations S
Output: A fine-tuned generation model p_θ

Load p_θ and train p_θ on $\mathcal{D}_L$
for $s = 1, \ldots, S$ **do**
  Initialize $\mathcal{D}_A = \emptyset$ and $\mathcal{D}_{L'} = \emptyset$
  // Synthetic Text Annotation
  for $\mathcal{A}_n \in \mathcal{D}_U$ **do**
    Generate $\boldsymbol{x}_n \sim p_\theta(\boldsymbol{x}_n \mid \mathcal{A}_n)$
    $\mathcal{D}_A \cup \{(\boldsymbol{x}_n, \mathcal{A}_n)\}$
  end for
  // Data Selection
  Compute threshold $\bar{\mu}$ and $\bar{s}$ using Eq. (6)
  for $(\boldsymbol{x}_n, \mathcal{A}_n) \in \mathcal{D}_A$ **do**
    if $\mathbb{E}[p_\theta] > \bar{\mu}$ and $Var[p_\theta] > \bar{s}$ **then**
      // Response Refinement
      Generate $\bar{\boldsymbol{x}}_n$ using Eq.(7)
      $\mathcal{D}_{L'} \cup \{(\bar{\boldsymbol{x}}_n, \mathcal{A}_n)\}$
    end if
  end for
  Fine-tune p_θ on $\mathcal{D}_L \cup \mathcal{D}_{L'}$
end for

3.2 Phase I: Uncertainty-based Data Selection

We hypothesize that the generation model is likely to gain little by learning from the data, if (1) it finds "too noisy", which may be out-of-domain or invalid; (2) it finds "too certain", which may be uninformative to learn from. Therefore, we propose to select the data which the current model finds "less noisy" and "more uncertain". Intuitively, data with "less noise" may provide helpful domain-specific knowledge to the model, meanwhile "more uncertainty" indicates the model has not learned well from the data yet, thus may produce incoherent responses.

Uncertainty estimation. We use the Monte Carlo Dropout method (Gal and Ghahramani, 2016; Mukherjee and Awadallah, 2020) to estimate the "noise" and "uncertainty" of each self-augmented data regarding the current model. For each self-augmented data $(\boldsymbol{x}, \mathcal{A})$, we enable dropouts before every hidden layer in the generation model, perform M forward passes through the model, and get M i.i.d. model likelihood estimations $\{p_{\theta_i}(\boldsymbol{x} \mid \mathcal{A})\}_{i=1}^{M}$. These M outputs are empirical samples of an approximated posterior distribution $p(\boldsymbol{x} \mid \mathcal{A})$ (Gal, 2016). Then, we compute the predictive mean $\mathbb{E}[p_\theta]$ of the approximated distribution

$p(\boldsymbol{x} \mid \mathcal{A})$ and predictive variance $Var[p_\theta]$ of the empirical samples:

$$\mathbb{E}[p_\theta] \approx \frac{1}{M}\sum_{i=1}^{M} p_{\theta_i}(\boldsymbol{x} \mid \mathcal{A}) \quad (4)$$

$$Var[p_\theta] \approx \frac{1}{M}\sum_{i=1}^{M} (p_{\theta_i}(\boldsymbol{x} \mid \mathcal{A}) - \mathbb{E}[p_\theta])^2 \quad (5)$$

A low predictive mean $\mathbb{E}[p_\theta]$ means the model finds the current data "too noisy", because it has a low likelihood estimation of the current data, which indicates the current data may be out-of-domain or invalid; while a low predictive variance $Var[p_\theta]$ means the model finds the current data "too certain", because all empirical samples have a similar likelihood estimation of the current data, which indicates the current data may be uninformative for the model to learn from. Therefore, we consider self-augmented data with both high predictive means and variances are examples of interest.

Selection strategy. The next question is *what are the thresholds for high predictive means and variances?* First, we calculate the corpus-level predictive mean μ_A of the self-augmented $\mathcal{D}_A$, and filter out the augmented data which have a lower predictive mean than μ_A, because we observe that such data are often very noisy and contain many redundant slots. Then, we combine and sort the original training data $\mathcal{D}_L$ and the remaining self-augmented data, and further remove the outliers (i.e. first and last 1% of datapoints). Assume that the collection of predictive mean scores $\mathbb{E}[p_\theta]$ and variance scores $Var[p_\theta]$ of the selected data follows a Gaussian distribution respectively, then the data selection threshold is defined as

$$\bar{\mu} = \frac{1}{N}\sum_{n=1}^{N} p_n, \quad \bar{s} = \frac{1}{N}\sum_{n=1}^{N} v_n \quad (6)$$

where p_n is the predictive mean and v_n is the predictive variance of the n-th selected data, N is the total number of original training data and remaining self-augmented data (after removing the outliers).

We select the self-augmented data with high $\mathbb{E}[p_\theta]$ (above the average predictive mean $\bar{\mu}$) and high $Var[p_\theta]$ (above the average predictive variance $\bar{s}$). We also explored other data selection strategies (detailed in §4.4), and find that selecting high $\mathbb{E}[p_\theta]$ and high $Var[p_\theta]$ data empirically brings more performance improvements than other strategies.

3.3 Phase II: Response Refinement

Since the large generation model is trained on a small training set, it is very likely to overfit and produce high-biased latent representations that cause the generation of inaccurate text responses. To reduce the risk of producing high-biased latent representations, we adopt dropout noise proposed in contrastive learning (Gao et al., 2021) into the latent representation during inference.

Specifically, for each selected input MR from **Phase I**, we enable the dropout masks of the model (placed on fully-connected layers as well as attention probabilities) at the decoding timestamp t, and compute R latent representations $\{\boldsymbol{h}^t_{\theta_i}\}_{i=1}^R$, then take an average over all latent representations to obtain the final latent representation for the current probability distribution:

$$p(\bar{x}_t \mid \bar{x}_{1:t-1}, \mathcal{A}) = \text{softmax}(\frac{1}{R}\sum_{i=1}^{R} \boldsymbol{h}^t_{\theta_i}) \quad (7)$$

Then, we generate the text response $\bar{\boldsymbol{x}}$ according to the probability distribution $p(\bar{x}_t \mid \bar{x}_{1:t-1}, \mathcal{A})$ and add the data $(\bar{\boldsymbol{x}}, \mathcal{A})$ into the pseudo-labeled dataset $\mathcal{D}_{L'}$. We fine-tune the generation model on both the original training set $\mathcal{D}_L$ and the pseudo-labeled dataset $\mathcal{D}_{L'}$. Fine-tuning the refined responses is shown to improve the model's final performances (detailed in §4.3).

4 Experiments

We conduct experiments to answer three research questions: (1) Is SA^2 self-training algorithm a helpful method to deal with the few-shot dialogue generation problem? (2) Can our data selection strategy effectively filter out the "too noisy" and "uninformative" augmented data? (3) Can our response refinement method help improve the performance of the NLG model?

4.1 Setups

Benchmark datasets. We evaluate our method on two few-shot dialogue generation benchmark datasets: FewShotWOZ (Peng et al., 2020) and FewShotSGD (Xu et al., 2021). FewShotWOZ has 7 domains and an average number of 50 training examples per domain. FewShotSGD has 16 domains and an average number of 35 training examples per domain. However, both datasets do not provide the development sets for hyper-parameter tuning. To create the standard training/dev/test data

	Restaurant		Laptop		Hotel		TV		Attraction		Train		Taxi	
	BLEU	ERR	BLEU	ERR	BLEU	ERR	BLEU	ERR	BLEU	ERR	BLEU	ERR	BLEU	ERR
SC-GPT	34.62	**1.95**	33.31	3.01	40.74	3.55	33.72	1.72	23.77	1.40	25.09	1.90	18.22	0.00
AUG-NLG	29.94	2.28	30.02	4.29	38.30	4.73	32.41	3.34	21.76	3.95	24.06	3.81	17.99	0.00
ST-ALL	33.84	6.51	34.40	4.28	39.68	1.78	34.88	1.76	24.32	3.19	24.47	3.87	17.89	0.00
ST-NLL	33.07	9.44	34.99	3.37	41.40	5.92	35.98	2.26	24.87	4.85	23.53	5.27	20.21	0.00
ST-SA^2 (ours)	**36.48**	2.60	**35.42**	**2.04**	**42.63**	**1.77**	**36.39**	**1.63**	**25.63**	**1.40**	**25.34**	**1.62**	**20.95**	**0.00**

Table 2: Automatic evaluation results on the test set of FewShotWOZ (BLEU↑, ERR↓). The results of **AUG-NLG** come from the data and code released by Xu et al. (2021), the other results come from our implementation.

	Restaurants	Hotels	Flights	Buses	Events	Rentalcars	Services	Ridesharing
SC-GPT	19.86	22.21	26.63	19.87	**26.41**	20.21	27.32	22.03
AUG-NLG	19.73	12.38	23.20	16.81	19.62	16.64	20.18	17.20
ST-ALL	19.71	21.45	26.90	19.76	25.68	20.22	27.59	21.14
ST-NLL	14.52	21.29	**27.59**	20.27	25.81	20.07	26.54	19.84
ST-SA^2 (ours)	**20.42**	**22.90**	27.12	**21.16**	25.32	**20.70**	**28.34**	**23.28**
	Movies	**Calendar**	**Banks**	**Music**	**Homes**	**Media**	**Travel**	**Weather**
SC-GPT	25.71	23.53	25.99	24.01	24.90	26.24	24.97	27.89
AUG-NLG	16.93	13.60	12.89	9.56	18.06	10.51	15.77	10.74
ST-ALL	26.19	24.86	25.03	24.62	24.97	26.56	25.28	28.06
ST-NLL	23.98	23.67	25.70	18.88	24.82	26.99	24.95	28.64
ST-SA^2 (ours)	**28.95**	**25.24**	**28.14**	**27.23**	**25.03**	**28.76**	**25.34**	**29.27**

Table 3: Automatic evaluation results of BLEU scores on the test set of FewShotSGD. The results of **AUG-NLG** come from the data and code released by Xu et al. (2021), the other results come from our implementation.

splits, we randomly sampled 10% data from the original test set as the dev set, and kept the training set unchanged. For fair comparisons across different methods, we evaluated all methods on the new split test set. The detailed data statistics of the two benchmarks are described in Appendix B.

Unlabeled data. The two benchmark datasets are sampled and constructed based on the three datasets: RNNLG (Wen et al., 2016), MultiWOZ (Budzianowski et al., 2018) and SGD (Rastogi et al., 2020b). To ensure the input MRs are within the same domain of the original training set $\mathcal{D}_L$, we collect all augmented MRs from the training set of RNNLG, MultiWOZ, and SGD. For FewShotWOZ, we collect an average number of 9,080 unlabeled MRs per domain. For FewShotSGD, we collect an average number of 7,532 unlabeled MRs per domain. The detailed data statistics of each domain are demonstrated in Appendix B.

Baselines. We compare our method with four baselines and describe the model configuration and training details in Appendix C. (1) **SC-GPT** (Peng et al., 2020) is the state-of-the-art pre-trained language model for NLG in task-oriented dialogue systems, which is further fine-tuned on each specific domain using the original training data $\mathcal{D}_L$; (2) **AUG-NLG** (Xu et al., 2021) leverages the pre-trained SC-GPT model, first trains it on its automatically retrieved augmented data, then fine-tunes it on each few-shot domain; (3) **ST-ALL** is the traditional self-training baseline which learns from all self-augmented data without any data selection and text refinement; (4) **ST-NLL** adopts the traditional self-training baseline but learns from the self-augmented data which has a lower than the average reconstruction loss according to the current generation model; (5) **ST-SA^2** is our method, in addition to our proposed data selection strategy and response refinement method, we apply a rule-based parser (Kedzie and McKeown, 2019) to heuristically filter out invalid responses that do not match the slot-value pairs in the input MRs on the FewShotWOZ dataset in order to achieve lower ERR.

Automatic evaluation. We follow the prior works (Wen et al., 2015; Peng et al., 2020; Xu et al., 2021) and use BLEU score and Slot Error Rate (ERR) for automatic evaluation. ERR is computed by exact matching the slot tokens in the generated responses as $ERR = (p + q)/N$, where N is the total number of slots in the MR, and p,q

	Restaurant		Laptop		Hotel		TV		Attraction		Train		Taxi	
	BLEU	ERR	BLEU	ERR	BLEU	ERR	BLEU	ERR	BLEU	ERR	BLEU	ERR	BLEU	ERR
ST-SA^2 (ours)	**36.48**	**2.60**	**35.42**	**2.04**	**42.63**	**1.77**	**36.39**	**1.63**	**25.63**	**1.40**	**25.34**	**1.62**	**20.95**	**0.00**
w/o aggregation	35.30	3.25	34.30	3.57	39.08	2.96	36.24	5.25	24.44	2.55	24.15	2.01	20.17	1.69
w/o filter	36.17	3.90	34.19	5.85	39.52	3.55	35.45	2.76	25.51	2.42	24.89	2.24	20.60	0.00

Table 4: Ablation study results on the test set of FewShotWOZ (BLEU↑, ERR↓).

	Informativeness ↑	Naturalness ↑
SC-GPT	2.62	2.32
ST-NLL	2.69	2.31
ST-SA^2 (ours)	**2.69**	**2.41**
Human	2.71	2.49

Table 5: Human evaluation results on the sampled test set of FewShotWOZ.

	Informativeness ↑	Naturalness ↑
SC-GPT	2.53	2.31
ST-ALL	2.55	2.40
ST-SA^2 (ours)	**2.69**	**2.42**
Human	2.69	2.56

Table 6: Human evaluation results on the sampled test set of FewShotSGD.

is the number of missing and redundant slots in the generated response. For each MR, we generate five responses and select the top one with the lowest ERR as the final output. Note that we only compute ERR on the FewShotWOZ dataset, because the FewShotSGD dataset does not release its evaluation script.

Human evaluation. We follow the prior works (Peng et al., 2020; Kale and Rastogi, 2020) and use Amazon Mechanical Turk to conduct human evaluation. We recruited master level workers with over 90% approval rate to compare and rate the responses generated by different methods and the the ground truth response. The workers are asked to rate the response on a scale of 1 (bad) to 3 (good) in terms of *informativeness* and *naturalness*. Informativeness indicates how much information from the input MR has been covered in the response, and naturalness measures whether the response looks coherent, grammatical, and natural. Each data pair is rated by 3 workers. We randomly sample 120 examples from each dataset, and collect a total of 2880 ratings.

4.2 Result Analysis

On FewShotWOZ. The automatic evaluation results in Table 2 show that **ST-SA^2** outperforms other baselines across all domains in both BLEU and ERR. Besides, we observe that **SC-GPT** is a strong baseline, and **ST-NLL** can bring more performance improvements than **AUG-NLG** and **ST-ALL** in 5 out of 7 domains, which shows the effectiveness of data selection in self-training. The human evaluation results in Table 5 indicate that **ST-SA^2** can generate more natural and informative responses than **SC-GPT** and **ST-NLL**. We provide some model generation results of different methods in Appendix E.

On FewShotSGD. The automatic evaluation results in Table 3 illustrate that **ST-SA^2** outperforms other baselines in 14 out of 16 domains in BLEU score. Additionally, we find that **ST-ALL** generally outperforms **AUG-NLG**, which indicates that additional pre-training on the retrieved task-relevant data does not necessarily help the model generate better responses. In contrast, the self-training method **ST-ALL** generally improves the model performances in 10 out of 16 domains, which shows the benefit of learning from self-augmented data. The human evaluation results in Table 6 demonstrate that **ST-SA^2** is capable to generate more informative and natural responses than **SC-GPT** and **ST-ALL**. We provide some model generation results of different methods in Appendix E.

4.3 Ablation Study on Response Refinement

To validate the effectiveness of the proposed response refinement method, we conduct ablation study on **ST-SA^2** by removing the representation aggregation in Equation 7 and the rule-based filter (Kedzie and McKeown, 2019) respectively. We observe from Table 4 that removing the representation aggregation during response refinement will lead to degraded performances in both BLEU and ERR across all domains, which indicates the importance of obtaining lower-biased latent representations during self-augmentation. Besides, we find that

	$\mathbb{E}[p_\theta]$	$Var[p_\theta]$	BLEU ↑	ERR ↓
1	low	low	32.72	1.62
2	low	high	32.24	1.62
3	high	low	33.18	2.28
4	high	high	**36.48**	2.60

Table 7: Different data selection strategy comparison of **ST-SA^2** in the **Restaurant** domain on the test set of FEWSHOTWOZ.

	Base Model	BLEU ↑	ERR ↓
1	GPT2	24.22	13.68
2	DialoGPT	14.77	20.84
3	SC-GPT	**36.48**	2.60

Table 8: Different base generation model comparison of **ST-SA^2** in the **Restaurant** domain on the test set of FEWSHOTWOZ.

removing the rule-based filter will lead to worse performances in ERR across all domains, which reveals that the model is still likely to generate incorrect responses, and those incorrect pseudo-labeled data will cause the model to learn irrelevant patterns and perform worse on the unseen test set.

4.4 Analysis of Other Components in SA^2 Self-training Algorithm

In this section, we provide additional empirical analysis on other components that will affect the performance of the SA^2 self-training algorithm, in order to gain more insights about the self-training technique in solving the few-shot NLG problem.

Data selection strategies. Table 7 compares different data selection strategies of **ST-SA^2** in the restaurant domain of FEWSHOTWOZ. We find that selecting low $\mathbb{E}[p_\theta]$ data will lead to degraded performance in BLEU score, because low $\mathbb{E}[p_\theta]$ data often contains more redundant tokens compared with the ground-truth response. Although low $\mathbb{E}[p_\theta]$ data gives lower ERR, the generated texts are not very natural and fluent. Selecting high $\mathbb{E}[p_\theta]$ and low $Var[p_\theta]$ data will also lead to degraded performance in the BLEU score, which is probably because the model overfits on the uninformative data. We provide some self-augmented and pseudo-labeled examples of different data selection strategies in Appendix D.

Base generation models. For the base generation model selection, we compare different pre-trained language models, including GPT2 (Radford et al., 2019), DialoGPT (Zhang et al., 2020) and SC-GPT. GPT2 is an open-end text generation model, and DialoGPT is an open-domain dialogue generation model. In contrast, SC-GPT is trained on around 400K MR-to-Text pairs in task-oriented dialogue generation datasets. As can be seen in Table 8, SC-GPT gives much better performance than GPT2 and DialoGPT, which indicates that selecting a suitable base generation model is critical for self-training.

	Epoch	LR	BLEU$_{dev}$ ↑	BLEU$_{test}$ ↑	ERR$_{test}$ ↓
1	10	1e-6	23.22	24.75	2.93
2	20	1e-6	22.96	24.63	2.04
3	20	5e-7	**23.43**	**25.63**	**1.40**
4	20	5e-8	23.29	24.82	1.91

Table 9: Different training hyper-parameters comparison of **ST-SA^2** in the **Attraction** domain of FEWSHOTWOZ, where **Epoch** is the number of training epochs within a self-training iteration, and **LR** is the initial learning rate at the beginning of each training epoch. We select the best model which has the highest **BLEU**$_{dev}$.

Training hyper-parameters. Table 9 compares different training hyper-parameters of **ST-SA^2** in the attraction domain of FEWSHOTWOZ dataset. We observe that the learning rate plays an essential role in training NLG models under the low-data setting. If the learning rate is too large, the development loss may not converge because the training set is too small; if the learning rate is too small, the model may get stuck into the local optimal. Finally, we find a good combination of learning rate and training epoch can help the model achieves the best performance, but the specific values vary across different domains. We provide training hyper-parameter configurations of each domain in Appendix C.

5 Conclusions

In this work, we present a two-phase self-augmentation self-training algorithm to deal with the few-shot dialogue generation problem in task-oriented dialogue systems. We propose to select informative input MRs based on model's prediction uncertainty, and improve the pseudo response generation by aggregating randomly perturbed latent representations. Empirical experiments on two few-shot NLG datasets show that our proposed method achieves the best performance among other baselines in both automatic and human evaluations.

Limitations

The performance of SA^2 self-training algorithm is influenced by the pre-trained language model used as the base generation model, because it offers the starting point for data selection and data augmentation. Building a good pre-trained language model for the MR-to-Text generation task is non-trivial, but future work in this direction will certainly benefit few-shot learning on dialogue generation. Besides, the SA^2 self-training algorithm requires large GPU resources for augmenting pseudo-labeled data. A more computationally efficient decoding method of Transformer-based models would save a significant amount of time and GPU resources.

Acknowledgments

The authors thank the anonymous reviewers for their useful comments and the UVa ILP group for helpful discussions. This work was supported by an Amazon Research Award to Yangfeng Ji.

References

Shreyan Bakshi, Soumya Batra, Peyman Heidari, Ankit Arun, Shashank Jain, and Michael White. 2021. Structure-to-text generation with self-training, acceptability classifiers and context-conditioning for the GEM shared task. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 136–147, Online. 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.

Ernie Chang, Xiaoyu Shen, Hui-Syuan Yeh, and Vera Demberg. 2021. On training instance selection for few-shot neural text 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 2: Short Papers)*, pages 8–13, Online. Association for Computational Linguistics.

Minmin Chen, Kilian Q Weinberger, and John Blitzer. 2011. Co-training for domain adaptation. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.

Zhiyu Chen, Harini Eavani, Wenhu Chen, Yinyin Liu, and William Yang Wang. 2020. Few-shot NLG with pre-trained language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 183–190, Online. Association for Computational Linguistics.

Adam Cheyer and Didier Guzzoni. 2006. Method and apparatus for building an intelligent automated assistant. *EPFL Scientific Publications*.

Alexander Fabbri, Simeng Han, Haoyuan Li, Haoran Li, Marjan Ghazvininejad, Shafiq Joty, Dragomir Radev, and Yashar Mehdad. 2021. Improving zero and few-shot abstractive summarization with intermediate fine-tuning and data augmentation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 704–717, Online. Association for Computational Linguistics.

Lingyun Feng, Minghui Qiu, Yaliang Li, Haitao Zheng, and Ying Shen. 2021. Wasserstein selective transfer learning for cross-domain text mining. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9772–9783, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Dongyeop Kang, Teruko Mitamura, and Eduard Hovy. 2020. GenAug: Data augmentation for finetuning text generators. In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 29–42, Online. Association for Computational Linguistics.

Yarin Gal. 2016. *Uncertainty in Deep Learning*. Ph.D. thesis, University of Cambridge.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA. PMLR.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2020. Revisiting self-training for neural sequence generation. In *International Conference on Learning Representations*.

Peyman Heidari, Arash Einolghozati, Shashank Jain, Soumya Batra, Lee Callender, Ankit Arun, Shawn Mei, Sonal Gupta, Pinar Donmez, Vikas Bhardwaj, Anuj Kumar, and Michael White. 2021. Getting to production with few-shot natural language generation models. In *Proceedings of the 22nd Annual*

Meeting of the Special Interest Group on Discourse and Dialogue, pages 66–76, Singapore and Online. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Mihir Kale and Abhinav Rastogi. 2020. Template guided text generation for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6505–6520, Online. Association for Computational Linguistics.

Chris Kedzie and Kathleen McKeown. 2019. A good sample is hard to find: Noise injection sampling and self-training for neural language generation models. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 584–593, Tokyo, Japan. Association for Computational Linguistics.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 704–710, Montreal, Quebec, Canada. Association for Computational Linguistics.

Yohan Lee. 2021. Improving end-to-end task-oriented dialog system with a simple auxiliary task. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1296–1303, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Sara Meftah, Nasredine Semmar, Youssef Tamaazousti, Hassane Essafi, and Fatiha Sadat. 2021. On the hidden negative transfer in sequential transfer learning for domain adaptation from news to tweets. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 140–145, Kyiv, Ukraine. Association for Computational Linguistics.

Sanket Vaibhav Mehta, Jinfeng Rao, Yi Tay, Mihir Kale, Ankur P. Parikh, and Emma Strubell. 2022. Improving compositional generalization with self-training for data-to-text generation.

Fei Mi, Wanhao Zhou, Lingjing Kong, Fengyu Cai, Minlie Huang, and Boi Faltings. 2021. Self-training improves pre-training for few-shot learning in task-oriented dialog systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1887–1898, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Subhabrata Mukherjee and Ahmed Awadallah. 2020. Uncertainty-aware self-training for few-shot text classification. In *Advances in Neural Information Processing Systems*, volume 33, pages 21199–21212. Curran Associates, Inc.

Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. 2020. Few-shot natural language generation for task-oriented dialog. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 172–182, Online. Association for Computational Linguistics.

Baolin Peng, Chenguang Zhu, Michael Zeng, and Jianfeng Gao. 2021. Data Augmentation for Spoken Language Understanding via Pretrained Language Models. In *Proc. Interspeech 2021*, pages 1219–1223.

Álvaro Peris and Francisco Casacuberta. 2018. Active learning for interactive neural machine translation of data streams. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 151–160, Brussels, Belgium. Association for Computational Linguistics.

Avinesh P.V.S and Christian M. Meyer. 2019. Data-efficient neural text compression with interactive learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2543–2554, Minneapolis, Minnesota. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

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

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020b. 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.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Zirui Wang, Zihang Dai, Barnabas Poczos, and Jaime Carbonell. 2019. Characterizing and avoiding negative transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

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.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. 2016. Multi-domain neural network language generation for spoken dialogue systems. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 120–129, San Diego, California. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721, Lisbon, Portugal. 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.

Xinnuo Xu, Guoyin Wang, Young-Bum Kim, and Sungjin Lee. 2021. AugNLG: Few-shot natural language generation using self-trained data augmentation. 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 1183–1195, Online. Association for Computational Linguistics.

Yunyi Yang, Yunhao Li, and Xiaojun Quan. 2021. Ubar: Towards fully end-to-end task-oriented dialog system with gpt-2. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14230–14238.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. 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.

Chenguang Zhu, Michael Zeng, and Xuedong Huang. 2019. Multi-task learning for natural language generation in task-oriented dialogue. 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 1261–1266, Hong Kong, China. Association for Computational Linguistics.

A Details of SA^2 Self-training Algorithm

We choose the pre-trained language model SC-GPT (Peng et al., 2020) as our base generation model p_θ. We collect in-domain MRs from the training set of existing task-oriented dialogue datasets, such as MultiWOZ corpus (Budzianowski et al., 2018) and Schema-Guided Dialog corpus (Rastogi et al., 2020a). We use nucleus sampling (Holtzman et al., 2020) with the threshold $p = 0.9$ to generate the output tokens for both synthetic text annotation and refined response generation.

B Dataset Details

Note that the original FEWSHOTWOZ and FEWSHOTSGD do not have a development set. To create the standard training/dev/test data splits, we randomly sampled 10% data from the original test set as the dev set, and kept the training set unchanged. For fair comparisons across different methods, we evaluated all methods on the newly split test set. The detailed data statistics of FEWSHOTWOZ is presented in Table 10. The detailed data statistics of FEWSHOTSGD is demonstrated in Table 11.

C Experimental Details

General Setups: The model is trained on an NVIDIA GeForce GTX 1080 Ti GPU server with 12GB memory. For the learning rate, we use the linear rate scheduler with no warm-ups. The AdamW optimizer (Loshchilov and Hutter, 2019) with default weight decay is used to update the parameters. For generation, we use nucleus sampling with $p = 0.9$ across all experiments.

SC-GPT: The pre-trained language model SC-GPT is loaded and fine-tuned on the original few-shot training set $\mathcal{D}_L$. The training epoch is set to 10, the batch size is set to 1, and the initial learning rate is set to 1e-5 across all domains in both FEWSHOTWOZ and FEWSHOTSGD.

AUG-NLG: There are two learning stages. In the first stage, the pre-trained language model SC-GPT is loaded and trained on the retrieved augmented data released by Xu et al. (2021), where the training epoch is set to 10, the batch size is set to 4, and the initial learning rate is set to 1e-5 across all domains in both datasets. In the second stage, the model checkpoint from the first stage is loaded and fine-tuned on the original few-shot training set $\mathcal{D}_L$, where the training epoch is set to 10, the batch size is set to 4, and the initial learning rate is set to 1e-5 across all domains in both datasets.

ST-ALL, ST-NLL, ST-SA^2: For all self-training methods, we start with the model checkpoint from the **SC-GPT** baseline. The maximum self-training iteration is set to $S = 5$. For evaluation, we save all model checkpoints at each self-training iteration, and report the best-performed model which has the highest **BLEU**$_{dev}$ score among all iterations (not necessarily the last iteration).

For **ST-ALL** and **ST-NLL**, in each self-training iteration, the training epoch is set to 10, the batch size is set to 4, and the initial learning rate is set to 1e-5 across all domains in both datasets.

For the model hyper-parameters in **ST-SA**^{2}, we set $M = 10$ in Equation 4 and Equation 5, and set $R = 10$ in Equation 7. For **ST-SA**^{2}, the training batch size is set to 4, and we report the detailed training epoch and initial learning rate across different domains and datasets for reproducibility purpose in Table 12 and Table 13.

D Self-Augmented Data Examples

Table 14 shows some examples of self-augmented data $\mathcal{D}_A$ and pseudo-labeled data $\mathcal{D}_{L'}$ under different data selection strategies in the **Restaurant** domain of FEWSHOTWOZ.

E Model Prediction Examples

Table 15 demonstrates some examples of model generation results in FEWSHOTSGD. Table 16 demonstrates some examples of model generation results in FEWSHOTWOZ.

	Restaurant	Laptop	Hotel	TV	Attraction	Train	Taxi
# Training Pairs	51	51	51	51	50	50	40
# Dev Pairs	12	137	7	68	34	65	4
# Test Pairs	117	1242	71	612	306	592	43
# Unlabeled Data	10,000	10,000	10,000	7,035	10,000	10,000	6,527

Table 10: Data statistics for the original manual-labeled data $\mathcal{D}_L$ and the unlabeled data $\mathcal{D}_U$ on FEWSHOTWOZ.

	Restaurants	Hotels	Flights	Buses	Events	Rentalcars	Services	Ridesharing
# Training Pairs	50	50	50	50	50	50	50	48
# Dev Pairs	961	401	272	427	836	287	793	819
# Test Pairs	8,657	3,615	2,453	3,845	7,526	2,592	7,146	7,378
# Unlabeled Data	10,000	10,000	10,000	10,000	10,000	10,000	10,000	8,259
	Movies	**Calendar**	**Banks**	**Music**	**Homes**	**Media**	**Travel**	**Weather**
# Training Pairs	30	25	23	21	21	14	14	11
# Dev Pairs	737	532	332	732	563	568	528	193
# Test Pairs	6,634	4,793	2,988	6,594	5,073	5,121	4,753	1,742
# Unlabeled Data	7,604	5,355	3,343	7,347	5,657	5,703	5,299	1,947

Table 11: Data statistics for the original manual-labeled data $\mathcal{D}_L$ and the unlabeled data $\mathcal{D}_U$ on FEWSHOTSGD.

	Domain	Epoch	LR	BLEU$_{dev}$	BLEU$_{test}$	ERR$_{test}$
1	Restaurant	10	8e-7	38.10	36.48	2.60
2	Laptop	10	5e-6	34.19	35.42	2.04
3	Hotel	10	1e-6	33.46	42.63	1.78
4	TV	10	1e-6	37.10	36.39	1.63
5	Attraction	20	5e-7	23.43	25.63	1.40
6	Train	10	8e-7	23.65	25.34	1.62
7	Taxi	10	1e-6	6.08	20.95	0.00

Table 12: Training hyper-parameter configurations of **ST-SA**^{2} in FEWSHOTWOZ, where **Epoch** is the number of training epochs within a self-training iteration, and **LR** is the initial learning rate at the beginning of each training epoch. We set the maximum self-training iteration $S = 5$, and select the model which has the highest **BLEU**$_{dev}$ across all self-training iterations.

	Domain	Epoch	LR	BLEU$_{dev}$	BLEU$_{test}$
1	Restaurants	10	1e-6	20.69	20.42
2	Hotels	10	1e-6	22.69	22.90
3	Flights	10	5e-6	25.82	27.12
4	Buses	10	1e-6	21.74	21.16
5	Events	10	5e-6	26.46	25.32
6	Rentalcars	10	1e-5	20.67	20.70
7	Services	10	1e-6	28.57	28.34
8	Ridesharing	10	1e-6	23.61	23.28
9	Movies	10	1e-6	29.37	28.95
10	Calendar	10	1e-6	25.97	25.24
11	Banks	10	1e-6	27.45	28.14
12	Music	10	1e-6	27.06	27.23
13	Homes	10	1e-6	24.45	25.03
14	Media	10	1e-5	28.40	28.76
15	Travel	10	1e-6	24.09	25.34
16	Weather	10	5e-7	27.43	29.27

Table 13: Training hyper-parameter configurations of **ST-SA**^{2} in FEWSHOTSGD, where **Epoch** is the number of training epochs within a self-training iteration, and **LR** is the initial learning rate at the beginning of each training epoch. We set the maximum self-training iteration $S = 5$, and select the model which has the highest **BLEU**$_{dev}$ across all self-training iterations.

low $\mathbb{E}[p_\theta]$ **low** $Var[p_\theta]$	
Input MR $\mathcal{D}_U$	*inform (choice = several) @ request (area = ?)*
Self-augmented data $\mathcal{D}_A$ (**Phase I**)	i have several restaurants that are good for lunch or dinner
Pseudo-labeled data $\mathcal{D}_{L'}$ (**Phase II**)	there are several restaurants that meet your needs
low $\mathbb{E}[p_\theta]$ **high** $Var[p_\theta]$	
Input MR $\mathcal{D}_U$	*inform (choice = several) @ request (area = ?)*
Self-augmented data $\mathcal{D}_A$ (**Phase I**)	there are several restaurants that match your criteria
Pseudo-labeled data $\mathcal{D}_{L'}$ (**Phase II**)	we have several restaurants that fit your criteria
high $\mathbb{E}[p_\theta]$ **low** $Var[p_\theta]$	
Input MR $\mathcal{D}_U$	*request (area = ?)*
Self-augmented data $\mathcal{D}_A$ (**Phase I**)	what is the area you looking for
Pseudo-labeled data $\mathcal{D}_{L'}$ (**Phase II**)	what is the area you looking for
high $\mathbb{E}[p_\theta]$ **high** $Var[p_\theta]$	
Input MR $\mathcal{D}_U$	*inform (choice = several) @ request (area = ?)*
Self-augmented data $\mathcal{D}_A$ (**Phase I**)	there are several restaurants in the area you'd like to dine in
Pseudo-labeled data $\mathcal{D}_{L'}$ (**Phase II**)	i have several restaurants that meet your criteria. what is the area you looking for

Table 14: Examples of self-augmented data $\mathcal{D}_A$ and pseudo-labeled data $\mathcal{D}_{L'}$ under different data selection strategies in the **Restaurant** domain of FEWSHOTWOZ.

Domain: Ridesharing	
Input MR	*INFORM (ride_fare = $11.02 ; approximate_ride_duration = 20)*
Human	it should take around 20 minutes and cost $11.02.
SC-GPT	the cost of the cab is $11.02 and the ride lasts 20 minutes.
AUG-NLG	the ride lasts 20 minutes and will cost $11.02.
ST-ALL	it will take 20 minutes and cost $11.02.
ST-NLL	the cost of the cab is $11.02 and the ride will last 20 minutes.
ST-SA^2 (ours)	the cost is $11.02 and it will take 20 minutes.
Domain: Media	
Input MR	*OFFER (title = the visitor, viva las vegas)*
Human	how about the visitor or viva las vega?
SC-GPT	the visitor, viva las vegas is the only movie that meets your criteria.
AUG-NLG	the visitor viva las vegas.
ST-ALL	the visitor, viva las vegas is also available.
ST-NLL	the visitor, viva las vegas is playing.
ST-SA^2 (ours)	how about the visitor, viva las vegas?
Domain: Hotels	
Input MR	*CONFIRM (where_to = sydney ; number_of_adults = 1 ; check_in_date = next monday ; check_out_date = march 11th)*
Human	please confirm: the location is sydney for a group of 1. the check in date will be next monday and the check out date will be march 11th.
SC-GPT	please confirm the following: you'd like to stay in sydney for 1 person from next monday to march 11th.
AUG-NLG	this is for a hotel in sydney. the reservation is for 1 person. the check-in date is next monday.
ST-ALL	sure. how about sydney? there's 1 person in your group, and you'll be arriving next monday. you'll be leaving march 11th.
ST-NLL	okay, i've got that. you 'd like a 1 person reservation for a house in sydney that 'd be held next monday, and will be'retired on march 11th. is that right?
ST-SA^2 (ours)	so to recap: you want a house in sydney for 1 person with a departure date of next monday and a march 11th check out.

Table 15: Examples of model generation results from the test set of FEWSHOTSGD.

Domain: Restaurant	
Input MR	*inform_no_match (pricerange = expensive ; area = bernal heights)*
Human	there are no expensive restaurants in bernal heights
SC-GPT	there are no expensive restaurants in the bernal heights area
AUG-NLG	unfortunately there is no restaurant near bernal heights that is expensive
ST-ALL	there is no restaurant in the expensive price range in the bernal heights
ST-NLL	unfortunately there is no restaurants in the expensive price range in the bernal heights area
ST-SA^2 (ours)	there are no expensive restaurants in the bernal heights area
Domain: Laptop	
Input MR	*inform_only_match (name = tecra morpheus 65 ; type = laptop ; batteryrating = exceptional ; pricerange = moderate)*
Human	the tecra morpheus 65 laptop is a moderate price range computer, with an exceptional battery rating
SC-GPT	tecra morpheus 65 laptop is an exceptional laptop in the moderate price range
AUG-NLG	tecra morpheus 65 is a nice laptop that is in the moderate price range. it has exceptional battery rating
ST-ALL	tecra morpheus 65 is a laptop in the moderate price range and comes with an exceptional battery rating
ST-NLL	the tecra morpheus 65 is a laptop in the moderate price range with exceptional battery rating
ST-SA^2 (ours)	tecra morpheus 65 is a laptop in the moderate price range with an exceptional battery rating
Domain: Hotel	
Input MR	*inform_count (type = hotel ; count = 182 ; hasinternet = dont_care)*
Human	there are 182 hotels if it does not matter if they have internet or not
SC-GPT	there are 182 hotels that do not care if there internet is
AUG-NLG	there are 182 hotels that do not care about internet access
ST-ALL	there are 182 hotels that do not care if they have internet or wi-fi
ST-NLL	there are 182 hotels that do not care if they have internet or not
ST-SA^2 (ours)	there are 182 hotels that do not care about internet
Domain: TV	
Input MR	*inform_no_match (type = television ; hasusbport = true ; screensizerange = medium ; family = l6)*
Human	i apologize but there are no televisions with medium screen size , that has usb ports
SC-GPT	no, there is no television in the l6 family with usb ports that is a medium screen size
AUG-NLG	the l6 television has usb ports, whereas the television without has only a medium screen size.
ST-ALL	there are no televisions that are in the medium family, with usb ports or in the l6 product family.
ST-NLL	there are no televisions with usb ports in the medium size family or l6 family.
ST-SA^2 (ours)	there are no televisions in the medium screen size range with usb ports in the l6 family.
Domain: Attraction	
Input MR	*sorry (area = centre ; choice = none)*
Human	i have none available in the centre. would you like a different area or attraction?
SC-GPT	i don't have any results for centre. would you like me to look in any other areas?
AUG-NLG	there aren't any listings for centre. would you like to try a specific area?
ST-ALL	there are none in the centre. would you like me to check other areas?
ST-NLL	there are none in the centre. would you like me to check in another area?
ST-SA^2 (ours)	sorry, i don't have any listings for centre. do you want me to check another area?
Domain: Train	
Input MR	*inform (leaveat = 16:34 ; id = tr8132) @ request (people = ?)*
Human	i have the tr8132 which leaves at 16:34. how many tickets please?
SC-GPT	tr8132 leaves at 16:34. how many tickets will you need?
AUG-NLG	there is a train that leaves at 16:34. how many tickets would you like?
ST-ALL	i have train tr8132 leaving at 16:34. how many tickets would you like?
ST-NLL	i have train tr8132 that leaves at 16:34. how many tickets would you like to book?
ST-SA^2 (ours)	tr8132 leaves at 16:34. how many tickets would you like?
Domain: Taxi	
Input MR	*request (destination = ? ; arriveby = ?)*
Human	i would be happy to book your taxi but i will need your destination and preferred arrival time as well.
SC-GPT	sure, where would you like to travel to? also where would you like to be picked up from?
AUG-NLG	what time would you like to arrive?
ST-ALL	sure, where would you like to go to and what time?
ST-NLL	what is your destination? do you have a arrival time in mind?
ST-SA^2 (ours)	sure, where would you like to go to? also, what time would you like to arrive?

Table 16: Examples of model generation results from the test set of FEWSHOTWOZ.