Improving Compositional Generalization with Self-Training for Data-to-Text Generation

Anonymous ACL submission

Abstract

Data-to-text generation focuses on generating fluent natural language responses from structured meaning representations (MRs). Such representations are compositional and it is costly to collect responses for all possible combinations of atomic meaning schemata, thereby necessitating few-shot generalization to novel MRs. In this work, we systematically study the compositional generalization of the state-of-the-art T5 models in few-shot data-to-text tasks. We show that T5 models fail to generalize to unseen MRs, and we propose a template-based input representation that considerably improves the model's generalization capability. To further improve the model's performance, we propose an approach based on self-training using fine-tuned BLEURT for pseudo-response selection. On the commonly-used SGD and Weather benchmarks, the proposed self-training approach improves tree accuracy by 46%+ and reduces the slot error rates by 73%+ over the strong T5 baselines in few-shot settings.

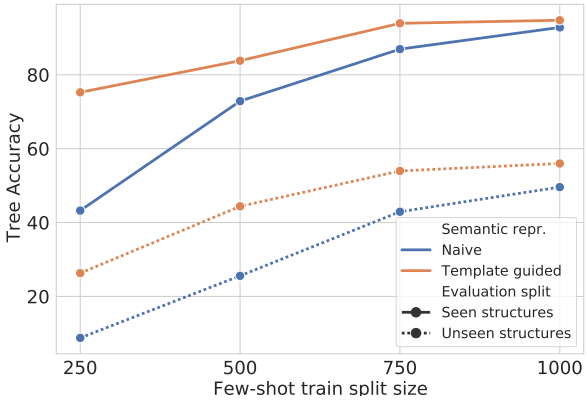

Figure 1: Performance comparison (tree accuracy) between different few-shot splits and semantic representations. T5-small undergoes a significant drop in performance on the unseen split and our template-guided representation improves generalization, reducing the gap.

1 Introduction

Data-to-text generation (Dušek et al., 2020; Shen et al., 2020) is a critical component in today's task-oriented dialog systems for producing fluent natural language responses to users' requests. The task takes structured meaning representations (MRs) as input for natural language text response generation. Such representations are compositional, which allows for the combination of atomic meaning units in various ways to express the rich semantics encoded in languages. Recently, large pre-trained language models (LMs) have shown impressive results on many language understanding and generation tasks (Howard and Ruder, 2018; Peters et al., 2018; Devlin et al., 2019; Raffel et al., 2020), however it remains unclear how well these LMs generalize compositionally to novel semantic representations.

There have been many studies revealing that large LMs often memorize the patterns from training data, while generalizing poorly to novel patterns. Compositionality in languages (Banarescu et al., 2013; Konstas et al., 2017) further aggravates such issues as the number of novel structural combinations exponentially increases with the number of atomic semantic units. In recent years, we have seen progress on benchmarking and measuring compositional generalization for languages (Andreas, 2019), from perspectives including specialized architectures (Lake, 2019; Rao et al., 2019) and learning strategies (Andreas, 2020; Akyürek et al., 2021). However, most of these works study the generalization for NLU tasks like question answering (Keysers et al., 2020) and semantic parsing (Kim and Linzen, 2020). To the best of our knowledge, compositional generalization for natural language generation is still an under-explored

Query: Is it jacket weather?

```
[DG_NO
]
[DG_INFORM
   [CONDITION light rain ]
   [HUMIDITY extremely humid ]
   [DATE_TIME [COLLOQUIAL today ] ]
   [LOCATION [CITY Palo Alto ]  ]
]
[DS_JUSTIFY
   [DG_RECOMMEND
      [ATTIRE_NOT jacket ]
      [LOCATION [CITY Palo Alto ] ]
      [DATE_TIME [COLLOQUIAL today ] ]
   ]
   [DG_INFORM
      [CONDITION_NOT cold ]
      [LOCATION [CITY Palo Alto ] ]
      [DATE_TIME [COLLOQUIAL today ] ]
   ]
]
```

(a) Naive Structured Input

Query: Is it jacket weather?

```
[DG_NO No
]
[DG_INFORM there will be
   [CONDITION light rain ]
   [HUMIDITY extremely humid ]
   [DATE_TIME at [COLLOQUIAL today ] ]
   [LOCATION in [CITY Palo Alto ]  ]
]
[DS_JUSTIFY
   [DG_RECOMMEND
      [ATTIRE_NOT jacket ] is not recommended
      [LOCATION in [CITY Palo Alto ] ]
      [DATE_TIME at [COLLOQUIAL today ] ]
   ] , because
   [DG_INFORM there won't be
      [CONDITION_NOT cold ]
      [LOCATION in [CITY Palo Alto ] ]
      [DATE_TIME at [COLLOQUIAL today ] ]
   ]
]
```

(b) Template Guided Structured Input

```
[DG_NO No
] ,
[DS_JUSTIFY
   [DG_RECOMMEND leave the
      [ATTIRE_NOT jacket ] at home
   ] because
   [DG_INFORM it isn't
      [CONDITION_NOT cold ]
      [DATE_TIME [COLLOQUIAL today ] ]
      [LOCATION in [CITY Palo Alto ] ]
   ] .
]
[DG_INFORM It'll be
   [HUMIDITY extremely humid ] with
   [CONDITION light rain ]
] .
```

Response: No, leave the jacket at home because it isn't cold today in Palo Alto. It'll be extremely humid with light rain.

(c) Structured Target Response

Figure 2: Example compositional meaning representations (discourse relations, dialog acts, arguments) (Balakrishnan et al., 2019) - (a) naive input, (b) template guided input, and (c) structurally annotated target response.

problem, which is the focus of this work.

To answer the question of whether pre-trained LMs still suffer from lack of compositional generalization, we start with an empirical evaluation of T5 (Raffel et al., 2020), the state-of-the-art model on data-to-text generation tasks (Kale and Rastogi, 2020b). In our study, we use the Weather dataset (Balakrishnan et al., 2019) consisting of tree-structured compositional MRs along with tree-structured output responses (see Figure 2 for (a) naive MR and (c) target response). For evaluation, we compute the tree accuracy (Balakrishnan et al., 2019) which measures exact match between input and generated tree-structures. In this study we observe 47%-80% (across different few-shot train splits) drop in the tree accuracy when evaluated on validation splits containing unseen tree-structures in comparison to splits containing seen tree-structures (Figure 1). Furthermore, simply increasing the model size from T5-*small* to T5-*large* does not close the generalization gap (Table 2), affirming our hypothesis that even strong seq-to-seq LMs fail to generalize compositionally.

Inspired by Kale and Rastogi (2020a), we examine whether template-guided MRs are effective over naive MRs for tackling compositional generalization in data-to-text tasks. We introduce a simple template engine that traverses the compositional MR in a top-down manner and converts it to a text representation (Figure 2(b)). We hypothesize that such a template-guided setup reduces the change in representation between LM pre-training and fine-tuning. With template-guided MRs, we report up to 2x increase in the tree accuracy over naive MRs on the validation split with unseen structures, demonstrating improved model generalization.

We also propose to self-train the generation model to further boost performance by mitigating data sparsity in the low-data regime without requiring additional manual annotation. Concretely, we augment the limited labeled MRs with unlabeled novel MRs to iteratively bootstrap the model. To filter out noisy pseudo responses during self-training, we repurpose BLEURT (Sellam et al., 2020), a learned metric, to be a quality estimator. We synthetically generate datasets for finetuning BLEURT with the goal of identifying hallucinations, missing slot-values, and ungrammatical responses. In sum, our overall approach improves the tree accuracy on unseen structures of the FewShotWeather dataset by 12.3%-46.4% over strong T5 baselines. On unseen schemata of the FewShotSGD dataset, we reduce the slot error rate by 54.4%-73.0%. We will release our code and data upon publication.

2 Case Study: Compositional Generalization in Data-to-Text Tasks

In this section, we are interested in investigating the following with respect to data-to-text tasks:

(Q1) Do current state-of-the-art generation models compositionally generalize?

(Q2) What is an effective semantic representation for tackling compositional generalization?

ID	Template Name	Template Body
1	DG_NO	[DG_NO No]
2	DS_JUSTIFY	[DS_JUSTIFY DG_RECOMMEND, because DG_INFORM]
3	DG_INFORM	*IsSet*($condition) ? DG_INFORM_CONDITION : DG_INFORM_CONDITION_NOT
4	DG_INFORM_CONDITION	[DG_INFORM there will be [CONDITION $condition] *Optional*([HUMIDITY $humidity]) DATETIME_AND_LOCATION]
5	DG_INFORM_CONDITION_NOT	[DG_INFORM there won't be [CONDITION $condition] DATETIME_AND_LOCATION]
6	DATETIME_AND_LOCATION	*Optional*(at [DATE_TIME $date_time]) *Optional*(in [LOCATION $location])
7	DG_RECOMMEND	[DG_Recommend [ATTIRE_NOT $attire] is not recommended DATETIME_AND_LOCATION]

Table 1: Example templates to convert a naive MR, Figure 2(a), to template guided text representation, Figure 2(b). A template could invoke other templates or some utility functions. The utility function *IsSet* denotes whether the argument is set, and function *Optional* returns empty text if the argument is not set.

(Q3) Does scaling model size (and training data) trivially solve compositional generalization?

Problem Setup Data-to-text generation is the task of generating natural language text y from meaning representation (MR) x. In the context of task-oriented dialog systems, the choice of MR ranges from a flat list of slot-value pairs (Dušek et al., 2018) to a more expressive tree structure. Balakrishnan et al. (2019) defines **tree-structured MRs** consisting of arguments, dialog acts, and discourse relations, which we use in this work. They report significant gains in the naturalness of the generated responses with tree-structured MRs on the Weather domain dataset. Figure 2 (a) visualizes an instantiation of such a tree-structured MR where the argument LOCATION is made up of a sub-argument (CITY), the dialog act RECOMMEND consists of three arguments (ATTIRE_NOT, LOCATION, DATE_TIME), and the discourse relation JUSTIFY captures the relationship between two dialog acts (RECOMMEND, INFORM).

We consider linearized versions of tree-structured MR x and output response y. Generating the tree structure in the output enables us to compute the tree accuracy which helps to assess the structural correctness of the predicted response.

FewShotWeather Dataset Due to the compositional nature of MRs, it is costly to collect responses for all combinations of discourse relations, dialog acts and arguments. In order to keep data labeling costs under control, we simulate a more realistic few-shot (or limited labeled data) setup. In the original Weather dataset, we have $25,390$ training examples spanning $4,690$ unique tree-structured MRs. An unique tree-structured MR is defined as a novel composition of discourse relations, dialog acts and argument names. Basically, they constitute non-terminals of a tree (Figure 2(a) without terminals or argument values like extremely humid, light rain, today, Palo Alto, jacket, and cold).

For the Weather dataset (Balakrishnan et al., 2019), we construct 4 few-shot splits: 1shot-250, 1shot-500, 1shot-750, and 1shot-1000, where 1shot-X denotes training split to include one example per unique tree-structured MR and in total X unique tree-structured MRs. Further, all X examples in 1shot-X are included while constructing 1shot-Y splits, where $X < Y$. We also make sure each discourse relation, dialog act and argument name is represented at least once in our few-shot splits. However, all combinations of these may not exist, thus allowing us to simulate structural shifts and evaluate compositional generalization. Based upon these splits, we construct two evaluation sets: *seen* tree-structures (overlapping with tree-structured MRs from 1shot-250) and *unseen* tree-structures (disjoint with tree-structured MRs from 1shot-1000) (see Section 4.1 for more details). Henceforth, all of the above splits constitute the *FewShotWeather* dataset. We will release these splits for future studies.

2.1 Semantic Representation

To answer (Q2), we use linearized tree structures as input to the T5 model (**naive representation**). However, T5 based models are pre-trained on normal text as input, thereby creating a representation discrepancy between pre-training and fine-tuning. To alleviate this discrepancy, we introduce a simple template engine that recursively traverses the compositional MR in a top-down manner to generate a structure-aware text representation (**template guided representation**). Some example templates

to convert naive representation (Figure 2(a)) to template guided representation (Figure 2(b)) are listed in Table 1. Each template, consisting of a name and a body, is invoked if a node in the MR (e.g., DG_INFORM) matches its name. A template can also invoke other templates or some utility functions. For example, template 3 could invoke templates 4 or 5 based on the returned value of the utility function *IsSet($condition)* (namely, whether the argument $condition is set or not). Such a template engine requires developing only a linear number of templates with respect to the number of meaning units to convert a compositional MR to a text representation, without writing a template for each unique MR (4,690 unique MRs in the dataset).

In our study, we fine-tune the T5-small model using different few-shot train splits and report tree accuracy on validation splits. We observe that current state-of-the-art generation models undergo a significant drop in performance when evaluated on unseen tree structures. Specifically, with naive input representation, we observe 47%-80% (across different few-shot train splits) drop in tree accuracy, thus, providing evidence to answer (Q1) that the current model does not generalize to novel MRs.

On experimentation with template guided MRs and 1shot-250 train split, the tree accuracy on validation unseen split increases from 8.77 to 26.3 (2x increase over naive MRs), thus, answering (Q2) favorably (Figure 1). However, across different few-shot train splits, template-guided MRs still undergo a significant 41%-65% drop in tree accuracy on the unseen split compared to the seen split.

2.2 Model scale

Recent studies (Kaplan et al., 2020; Tay et al., 2021) show that model scale can affect the performance on several pre-training and downstream tasks. To understand how model scale affects the generalization to unseen structures, we consider three T5 variants: T5-small (77M), T5-base (120M), and T5-large (800M). We fine-tune each of these models on the full training data (16,816 examples corresponding to 1000 unique tree-structured MRs from 1shot-1000 split) and convincingly answer (Q3): Increasing the model (and dataset) size does not close the performance gap between seen and unseen splits (Table 2). Surprisingly, we observe that the T5-small model performs similarly or better than its larger counterparts. We use T5-small for the remaining experiments.

Model Size	Val. Seen	Val. Unseen
T5-small (77M)	99.54	64.02
T5-base (120M)	99.63	55.80
T5-large (800M)	99.36	58.45

Table 2: Performance comparison (tree accuracy) between different T5 model variants. Each T5 model is fine-tuned on full Weather dataset (16,816 examples) and evaluated on validation seen and unseen splits. We observe that increasing the model size does not close the compositional generalization gap.

3 Self-training

As discussed earlier, the compositional nature of MRs makes it difficult to collect responses for all combinations. However, with access to data simulators (Rastogi et al., 2020), it is feasible to automatically generate large amounts of unlabeled MRs. Given limited labeled MRs, $S = \{x^i, y^i\}_{i=1}^n$, and assuming access to unlabeled MRs, $U = \{x^i\}_{i=1}^m$, we investigate self-training (Scudder, 1965), a semi-supervised learning approach to effectively use U to improve compositional generalization.

Self-training starts from a model trained on labeled data S, iteratively applies the current model to generate pseudo-labels on unlabeled data U, and then re-trains the current model on the augmented version of S and (subset of) U. For self-training to be effective, one needs to carefully select confident pseudo labels to alleviate the risk of reinforcing the model's mistakes (He et al., 2020). This issue gets further exacerbated in the context of generation tasks, where neural models are prone to hallucinate additional content not supported by the input (Maynez et al., 2020).

With recent developments in learned evaluation metrics that penalize the model for hallucination, fluency, etc., we pose the question: *Can we repurpose those metrics to assess the quality of pseudo-responses during self-training?* Formally, given a pair of template guided MR (source) and model predicted response (candidate), we want a model that estimates the response quality by looking for hallucinations, fluency, coverage of argument value-pairs. Ideally, to learn such a model we require a large amount of positive and negative text pairs. To alleviate this requirement, we propose synthesizing the examples using the limited labeled task dataset. Furthermore, we initialize our quality estimation model using a pre-trained BLEURT (Sellam et al., 2020), which is shown to be sample efficient and robust to data shifts as a learned evaluation metric.

Soruce (text-to-text input): there will be light freezing fog with a temperature high of 74 low of 61 at next friday

Positive candidate (target response): next friday will have a high of 74 , a low of 61 , and a light freezing fog

Negative candidates:

[retrieving similar examples] next friday will be cloudy with a high of 74 , a low of 61 , and thunderstorms and rain

[pairing with reference] there will be light freezing fog with a temperature high of 74 low of 61 at next friday

[swapping words] next friday ~~will~~ of have a high ~~of~~ will 74 , a low of 61 , and a light freezing fog

[repeating phrases] next friday will have a high of 74 , a low of 61 of 61 , and a light freezing fog

[dropping phrases] next friday will have a high of 74 , ~~a low of 61 ,~~ and a light freezing fog

[flipping digits] next friday will have a high of 78 , a low of 61 , and a light freezing fog

Figure 3: Synthetically constructed positive and negative candidates for BLEURT fine-tuning.

Once we have a fine-tuned BLEURT model, we use it to select pseudo-responses using a selection threshold for self-training.

3.1 Fine-tuning BLEURT

We synthetically generate the dataset for fine-tuning BLEURT using the labeled dataset available for each of our experiments. Template guided inputs and ground truth target responses are paired as positive examples (rating: 1.0). We use the following transformations on the target responses to create negative examples (rating: 0.0):

Retrieving similar examples: For every input x, we rank all other inputs from the dataset using the BLEU score and select top-k examples below a certain threshold (90.0). Target responses corresponding to these top-k examples are paired with x to construct negative examples. Intuitively, these responses partially overlap with input x in terms of the content and inform a fine-tuned model to handle hallucinations.

Pairing with reference: Template guided inputs need not be grammatically correct. Pairing the input x with itself as a response provides grammatically incorrect negative examples.

Swapping, repeating and dropping phrases, flipping digits: Using these methods, we prepare a fine-tuned BLEURT for structurally inconsistent behaviors of the NLG system. Figure 3 visualizes an instantiation of different transformations to construct negative examples.

4 Experimentation

4.1 Datasets and Metrics

FewShotWeather The original Weather dataset (Balakrishnan et al., 2019) has $25,390$ training examples. Each example consists of a user query, the tree-structured MR, the tree-structured annotated response and metadata. As discussed in Section 2, we create new canonical subsets for compositional generalization experiments, FewShotWeather with 1shot-250 (approx. 1% of original training data), 1shot-500, 1shot-750, and 1shot-1000 splits. We repurpose all the remaining 24k training examples as unlabeled examples for self-training. Our evaluation splits have $1,087/1,121$ (val/test) examples with seen tree-structures, and $1,095/1,170$ (val/test) examples with novel tree-structures. We report tree accuracy and BLEU-4 (Papineni et al., 2002) for the FewShotWeather dataset.

FewShotSGD The original multi-domain Schema Guided Dialogue (SGD) dataset (Rastogi et al., 2020) has 160k examples spanning across 20 domains (e.g., Banks, Travel, Weather, etc.). For each of these domains, there are different services with a total of 45 different schemata. Schema here refers to the combination of intents and slots, which change with services and domains. Further, not all domains and services are observed during training. Therefore, we use this dataset to study generalization to unseen schemata. Specifically, we use the few-shot variant of the dataset, FewShotSGD, as introduced by Kale and Rastogi (2020a). The FewShotSGD benchmark consists of k-shot splits (5/10/20/40), where k denotes the number of dialogues selected per train domain. The few-shot train splits have 558/1,075/2,140/4,312 (5/10/20/40-shot) examples. Evaluation splits have 13,748/10,216 (val/test) examples with seen schema, and 10,386/26,568 (val/test) examples with novel schema. Following Kale and Rastogi (2020a), we report BLEU-4 and slot error rate (SER) (Dušek and Jurcicek, 2019). SER measures the fraction of examples where at least one slot was incorrectly copied from the input (lower SER is better).

Pseudo-response selection strategy	FewShotWeather Train split	Seen structures BLEU ↑	Seen structures Tree Acc. ↑	Unseen structures BLEU ↑	Unseen structures Tree Acc. ↑	FewShotSGD Train split	Seen schemata BLEU ↑	Seen schemata SER ↓	Unseen schemata BLEU ↑	Unseen schemata SER ↓
None	1shot-250	69.16	73.68	50.40	29.83	5-shot (558)	20.66	22.84	20.52	19.93
Vanilla		69.25	73.77	51.87	31.37		23.03	15.15	21.97	15.96
BLEURT		**69.59**	**84.12**	**52.34**	**43.68**		**25.22**	**4.78**	**24.13**	**5.39**
None	1shot-500	69.40	83.59	53.62	46.58	10-shot (1,075)	21.45	21.64	22.79	14.98
Vanilla		68.75	89.21	54.27	49.91		23.50	17.90	24.38	7.67
BLEURT		68.19	**93.40**	**56.12**	**55.30**		**25.63**	**4.29**	**25.49**	**3.82**
None	1shot-750	69.81	92.86	54.49	54.02	20-shot (2,140)	22.84	16.74	25.14	11.51
Vanilla		73.02	96.61	54.32	54.19		23.19	14.92	25.47	9.11
BLEURT		72.00	**97.23**	**55.21**	**58.89**		**26.63**	**3.33**	**27.38**	**3.77**
None	1shot-1000	72.89	95.18	53.97	55.64	40-shot (4,312)	25.72	7.60	26.52	5.97
Vanilla		73.38	96.16	55.04	60.09		26.65	5.00	26.61	4.20
BLEURT		**73.82**	**98.48**	**57.11**	**62.48**		**27.48**	**2.37**	**27.53**	**2.72**
Full	16,816	74.43	99.55	62.44	65.47	164,978	29.28	1.12	28.76	1.54

Table 3: Comparing performance in terms of BLEU, tree accuracy (Tree Acc.), and slot error rate (SER) between vanilla and BLEURT based pseudo-response selection strategies on FewShotWeather and FewShotSGD test splits. All results are for the T5-small model with template guided input representation. Pseudo-response selection strategy *None* denotes fine-tuned T5-small baseline without self-training. ↑ indicates higher is better, ↓ indicates lower is better. Overall, BLEURT based self-training improves the performance on (un)seen structures/ (un)seen schemata over vanilla self-training.

4.2 Implementation

For each of the experiments we fine-tune the off-the shelf T5.1.1.small checkpoint (Raffel et al., 2020). It has 6 layers each in encoder and decoder with a total of 77M parameters. We set the max sequence length to 512, batch size to 16 and a constant learning rate of 0.001 for Adafactor optimizer (Shazeer and Stern, 2018). We fine-tune the model for 5k steps, evaluate after every 200 steps and retain the checkpoint yielding best tree accuracy (for FewShotWeather) or BLEU (for FewShotSGD) on the held-out validation seen split. During inference, we set the beam size to 4 and length penalty $\alpha = 0.6$.

While constructing the fine-tuning dataset for BLEURT, we generate up to 4 different negative candidates corresponding to each of the 6 transformations. We upsample the positive examples to be half the total number of negative examples and retain random 10% of total examples for validation set. For fine-tuning the BLEURT model, we start with publicly available BLEURT-20-D12 (Sellam et al., 2020). We set the max sequence length to 512, batch size to 32, a learning rate 1e-6, and fine-tune for 100k steps. We use the held-out validation set to select the best checkpoint for self-training.

4.3 Self-Training

In this section, we compare the performance of BLEURT based pseudo-response selection strategy with that of *vanilla* self-training. For each experiment, we *randomly* sample an equal number of examples for vanilla self-training and the BLEURT model to explicitly control for the sample complexity. We run 3 iterations of the self-training unless explicitly specified and set the BLEURT score selection threshold to 0.99. We study the performance on a dataset (FewShotWeather) with tree-structured outputs as well as show the generality of our method on a dataset (FewShotSGD) without explicit tree-structured outputs.

Unseen tree structures (FewShotWeather) Table 3 reports the performance of different methods as a function of the number of labeled examples. We observe that the performance for all methods improves with more training data. Across all few-shot splits, we observe that BLEURT based self-training improves over vanilla self-training both in terms of tree accuracy and BLEU. Empirically, we see that relative gains in tree accuracy (over the T5-small baseline) from vanilla self-training are comparable on both unseen and seen splits (e.g., 7.15% v.s. 6.72%, 1shot-500). On the other hand, BLEURT based self-training significantly improves the relative performance on the unseen split in comparison to seen splits (e.g., 18.72% vs. 10.5%, 1shot-500), thus showcasing the effectiveness of selecting quality pseudo-responses for improving performance on unseen tree-structures.

Model	Self-training iteration	No. of training examples	FewShotWeather			
			Seen structures		Unseen structures	
			BLEU ↑	Tree Acc. ↑	BLEU ↑	Tree Acc. ↑
Baseline	-	250	69.16	73.68	50.40	29.83
Vanilla	1	+ 14,742	69.25	73.77	51.87	31.37
	2	+ 4,170	68.72	73.06	51.92	31.11
BLEURT-250	1	+ 14,742	69.64	83.85	52.10	41.03
	2	+ 4,170	69.59	84.12	52.34	43.68
BLEURT-1000	1	+ 14,021	70.95	84.83	52.13	45.47
	2	+ 4,772	**70.47**	**85.64**	**53.08**	**47.44**

Table 4: Model performance over multiple self-training iterations with FewShotWeather 1shot-250 train split. BLEURT-X denotes BLEURT model fine-tuned using 1shot-X train split. We observe that BLEURT model fine-tuned with larger datasets further enhances the self-training performance, especially on unseen structures.

Unseen schema (FewShotSGD) Table 3 reports the performance on the FewShotSGD dataset. Similar to results on the FewShotWeather dataset, we observe that the performance improves with more training data. Further, the performance of the baseline T5-small model is comparable to seen and unseen schemata. These gains can be attributed to the benefits of using template guided MRs. In comparison to vanilla self-training, BLEURT based approach improves the overall performance across all few-shot splits on both seen and unseen schema. For example, with 5-shot experiments, BLEURT based selection strategy reduces the SER on unseen schema from 19.93 to 5.39 (73% improvement) in comparison to the baseline T5 model. On the other hand, vanilla self-training reduces the SER only by 3.97 (20%), thus showcasing the effectiveness of the proposed approach in filtering pseudo-responses with missing slot-value pairs. These results confirm that BLEURT based self-training is a generic method and can be plugged in to existing methods to improve the few-shot generalization capabilities of existing SOTA generation models.

Performance with respect to self-training iterations We iteratively self-train the model starting from a T5-small baseline and continue adding unlabeled examples up to 3 iterations. From Table 4 and 6, we see that model performance improves across the self-training iterations. However, the number of additional examples added decreases over iterations, thus suggesting that 2-3 iterations might be enough to obtain benefits from self-training.

Quality of fine-tuned BLEURT models For all our experiments, we use the few-shot labeled datasets for fine-tuning the BLEURT model. To investigate self-training performance with a BLEURT model fine-tuned on a large dataset, we set up an experiment on the FewShotWeather dataset, where we fine-tune the BLEURT model on a 1shot-1000 train split (BLEURT-1000) and use it for self-training with 1shot-250. From Table 4, we see that self-training with BLEURT-1000 performs significantly better than BLEURT-250, especially on unseen structures, thereby confirming the intuition that self-training is sensitive to the quality of the BLEURT model.

4.4 Human evaluation

Aside from automatic metrics-based evaluation, we also perform a human evaluation study by asking annotators to assess the quality of the generated responses from different models. For each example, human annotators are shown user query, generated response and the ground truth response. They are asked to provide ratings on a scale of 1 (bad), 2 (slightly bad) to 3 (good) along three dimensions: *grammaticality*, *naturalness*, *informativeness*, and binary rating for *accuracy*. Similar to (Balakrishnan et al., 2019), grammaticality evaluates the response for subject-verb agreement, repetitions, and grammatical completeness. Naturalness measures whether the response sounds coherent and natural by the response itself. Informativeness measures whether the response contains the right amount of relevant information to the user query and accuracy evaluates the response for hallucinations (incorrectly added slots), missing slots by comparing it against the reference. For each evaluation split (seen/unseen), we randomly select 200 examples and collect ratings from 3 different annotators. For the FewShotWeather/SGD datasets, we consider models trained with 1shot-250/5-shot splits and compare them with models fine-tuned on the full dataset. In total, we collect 7,200 annotations,

Model	Gram	Nat	Info	Acc
FewShotWeather (Seen split)				
Baseline	2.59	2.55	2.05	0.94
BLEURT	**2.66**^{1}	**2.63**^{1}	2.06	0.93
Full	**2.66**^{1}	2.61	**2.07**	**0.95**
FewShotWeather (Unseen split)				
Baseline	2.43	2.41	1.96	0.79
BLEURT	2.50^{1}	2.46^{1}	**1.99**	0.80
Full	**2.53**^{1}	**2.50**^{1}	1.98	**0.86**^{1,2}
FewShotSGD (Seen split)				
Baseline	2.72	2.66^{2}	1.93	0.76
BLEURT	2.69	2.59	**2.00**^{1}	0.88^{1}
Full	**2.83**^{1,2}	**2.74**^{1,2}	**2.00**^{1}	**0.94**^{1,2}
FewShotSGD (Unseen split)				
Baseline	2.70	2.61	1.93	0.72
BLEURT	2.67	2.60	1.98^{1}	0.86^{1}
Full	**2.83**^{1,2}	**2.73**^{1,2}	**2.02**^{1,2}	**0.94**^{1,2}

Table 5: Human evaluation results comparing different models. Grammaticality (Gram), naturalness (Nat), and informativeness (Info) are on the scale of 1 to 3 and accuracy (Acc) is binary. The superscripts $1, 2, 3$ indicate that model is significantly better than baseline, BLEURT-based self-training, and model trained with full data, respectively, as determined by one-sided paired t-test with $p < 0.05$.

each with 3 ratings. Table 5 reports results for human evaluation study and Table 7, 8 (Appendix A.1) visualize the sample responses generated using different models on unseen splits.

FewShotWeather Similar to automatic metrics, we see a drop in human ratings on the unseen split (compared to seen split), confirming the model's lack of generalization to novel MRs. On both the evaluation splits, our approach outperforms the baseline model with significant results on grammaticality and naturalness ratings. Moreover, the responses from the self-trained model are comparable (in terms of the human ratings) with that of the model fine-tuned with the full dataset, demonstrating the effectiveness of our approach.

FewShotSGD Apart from generating natural responses, model responses must be factually grounded in the input data and address user queries. On FewShotSGD, we see that our approach significantly improves informativeness and accuracy rating over the baseline model. Surprisingly, we see a drop on naturalness when evaluated on seen schemata.

5 Related Work

Data-to-Text Generation While early research focused on rule-based methods (Reiter and Dale, 2000), more recent work has relied heavily on neural methods (Wen et al., 2015; Marcheggiani and Perez-Beltrachini, 2018). Some recent works (Kale and Rastogi (2020b), Peng et al. (2020), Kale and Roy (2020)) showed that transfer learning from pre-trained language models can improve generalization capabilities and sample efficiency. In other lines of work, Ferreira et al. (2019); Moryossef et al. (2019) find that pipelined neural approaches with explicit planning steps can outperform their end-to-end counterparts, while Kale and Rastogi (2020a) and Du et al. (2020) showed the benefits of schema and template guided input representations. Inspired by (Kale and Rastogi, 2020a) we propose a simple and generic way to produce text-to-text representation, and study how it impacts compositional generalization.

Self-training for NLG (He et al., 2020) revisits the problem of self-training for NLG. They found that noise (from perturbing the input space) helps in self-training and propose a "noisy" version of self-training by augmenting vanilla training with the inputs from a reconstruction model. Building on this idea, the contemporary work (Heidari et al., 2021) on few-shot data-to-text generation proposes to self-train the model and shows efficacy on the Weather dataset. Another contemporary work (Li et al., 2021) proposes to use constrained decoding to generate valid pseudo-responses for self-training and show convincing benefits. However, our work focuses on compositional generalization, rather than the pure few-shot learning setup. We propose a BLEURT-based self-training method, which is more generic than pseudo-response selection methods that rely on output structures.

6 Conclusion

We systematically study the problem of compositional generalization for data-to-text generation and show that existing state-of-the-art generation models do not generalize to unseen structures. We propose a simple and generic way to produce template guided text representation for response generation, and demonstrate its effectiveness on both seen and unseen structures. Further, we introduce a generic self-training approach that leverages fine-tuned BLEURT for pseudo response selection and show significant improvements over vanilla self-training on existing few-shot data-to-text generation benchmarks.

References

Ekin Akyürek, Afra Feyza Akyürek, and Jacob Andreas. 2021. Learning to recombine and resample data for compositional generalization. In *International Conference on Learning Representations*.

Jacob Andreas. 2019. Measuring compositionality in representation learning. In *International Conference on Learning Representations*.

Jacob Andreas. 2020. Good-enough compositional data augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7556–7566.

Anusha Balakrishnan, Jinfeng Rao, Kartikeya Upasani, Michael White, and Rajen Subba. 2019. Constrained decoding for neural nlg from compositional representations in task-oriented dialogue. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 831–844.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

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

Yuheng Du, Shereen Oraby, Vittorio Perera, Minmin Shen, Anjali Narayan-Chen, Tagyoung Chung, Anushree Venkatesh, and Dilek Hakkani-Tur. 2020. Schema-guided natural language generation. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 283–295.

Ondřej Dušek and Filip Jurcicek. 2019. Neural generation for czech: Data and baselines. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 563–574.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2018. Findings of the e2e nlg challenge. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 322–328.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge. *Computer Speech & Language*, 59:123–156.

Thiago Castro Ferreira, Chris van der Lee, Emiel van Miltenburg, and Emiel Krahmer. 2019. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. 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 552–562.

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, et al. 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.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339.

Mihir Kale and Abhinav Rastogi. 2020a. 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.

Mihir Kale and Abhinav Rastogi. 2020b. Text-to-text pre-training for data-to-text tasks. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 97–102.

Mihir Kale and Scott Roy. 2020. Machine translation pre-training for data-to-text generation–a case study in czech. *arXiv preprint arXiv:2004.02077*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, et al. 2020. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*.

Najoung Kim and Tal Linzen. 2020. Cogs: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105.

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural amr: Sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–157.

Brenden M Lake. 2019. Compositional generalization through meta sequence-to-sequence learning. *Advances in Neural Information Processing Systems*, 32:9791–9801.

Xintong Li, Symon Stevens-Guille, Aleksandre Maskharashvili, and Michael White. 2021. Self-training for compositional neural nlg in task-oriented dialogue. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 87–102.

Diego Marcheggiani and Laura Perez-Beltrachini. 2018. Deep graph convolutional encoders for structured data to text generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 1–9.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. Step-by-step: Separating planning from realization in neural data-to-text generation. 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 2267–2277.

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.

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 *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 172–182.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL-HLT*.

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

Jinfeng Rao, Kartikeya Upasani, Anusha Balakrishnan, Michael White, Anuj Kumar, and Rajen Subba. 2019. A tree-to-sequence model for neural nlg in task-oriented dialog. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 95–100.

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

Ehud Reiter and Robert Dale. 2000. *Building natural language generation systems*. Cambridge university press.

Henry Scudder. 1965. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371.

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.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

Xiaoyu Shen, Ernie Chang, Hui Su, Cheng Niu, and Dietrich Klakow. 2020. Neural data-to-text generation via jointly learning the segmentation and correspondence. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7155–7165.

Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. 2021. Scale efficiently: Insights from pre-training and fine-tuning transformers. *arXiv preprint arXiv:2109.10686*.

TH Wen, M Gašić, N Mrkšić, PH Su, D Vandyke, and S Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Conference Proceedings-EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721.

A Appendix

A.1 Qualitative Analysis

In Table 7, 8 we visualize the sample responses generated using different models for unseen test splits. We consider three models: T5-small baseline, BLEURT based self-training, and model trained with full data. For the FewShotWeather/ FewShotSGD datasets, we consider models trained with 1shot-250/ 5-shot train splits. We see that the baseline model fails to generate responses that are coherent and factually grounded in the input. They hallucinate to generate novel concepts like *cloudy hail*, drop relevant details like *cafe located in Emeryville*, and are repetitive in nature. We

Model	Self-training iteration	No. of training examples	FewShotSGD			
			Seen schemata		Unseen schemata	
			BLEU ↑	SER ↓	BLEU ↑	SER ↓
Baseline	-	558	20.66	22.84	20.52	19.93
Vanilla	1	+ 101,577	22.96	16.26	21.69	15.19
	2	+ 30,867	22.94	15.43	21.94	16.04
	3	+ 5,998	23.03	15.15	21.97	15.96
BLEURT	1	+ 101,577	24.34	9.85	23.29	8.43
	2	+ 30,867	24.84	6.96	23.64	6.58
	3	+ 5,998	25.22	4.78	24.13	5.39

Table 6: Model performance over multiple self-training iterations with 5-shot train split (FewShotSGD). ↑ indicates higher is better, ↓ indicates lower is better. We observe that model performance increases with the self-training iteration. However, the number of additional examples added decreases over iteration, suggesting that 2-3 iterations are sufficient for self-training.

also report the BLEURT scores and see that it is reflective of the response quality.

Fields	BLEURT	Input or output response
User query	-	Will it be raining at 2 pm today
Template	-	No, and there won't be rain from 03:00 PM to 07:00 PM there will be partly cloudy with a temperature of 80 from 03:00 PM to 07:00 PM
Reference	-	No, rain isn't expected at 03:00 PM. Instead, expect partly cloudy skies and 80 fahrenheit continuing through 07:00 PM.
Predictions		
Baseline	0.001	No, it won't rain this today from 03:00 PM to 07:00 PM and it will be 80 fahrenheit. Between 03:00 PM and 07:00 PM, you can expect partly cloudy skies, and 80 fahrenheit
Self-training	0.997	No, it won't rain today from 03:00 PM to 07:00 PM. Between 03:00 PM and 07:00 PM, you can expect partly cloudy skies, and 80 fahrenheit
Full	0.999	No, it won't rain from 03:00 PM until 07:00 PM. It'll be 80 and partly cloudy
User query	-	What will the temperature be in Birmingham today?
Template	-	There will be with a temperature high of 69 low of 40 at today in Birmingham England United Kingdom
Reference	-	The high for today will be 69 and the low will be 40 fahrenheit in Birmingham, England, United Kingdom
Predictions		
Baseline	0.000	The temperatures in Birmingham, England, United Kingdom in Birmingham, England, United Kingdom range between 40 and 69
Self-training	0.990	The temperatures in Birmingham, England, United Kingdom today range between 40 and 69
Full	0.999	There will be a high of 69 and low of 40 today in Birmingham, England, United Kingdom
User query	-	is there hail today
Template	-	No, but there will be heavy freezing rain and cloudy at today in Oxford
Reference	-	No, today in Oxford is cloudy with heavy freezing rain
Predictions		
Baseline	-0.001	No, but there is cloudy hail today in Oxford
Self-training	0.999	No, but there is cloudy weather today with heavy freezing rain in Oxford
Full	0.998	No, but it'll be cloudy with heavy freezing rain today in Oxford
User query	-	Should I wear a heavy jacket wednesday
Template	-	No there will be mostly sunny with a temperature high of 66 low of 47 at next Wednesday jacket is not recommended at next Wednesday, because there won't be cold at next Wednesday
Reference	-	No, you don't need a jacket, because it won't be cold next Wednesday. Expect a high of 66 and a low of 47, with mostly sunny skies
Predictions		
Baseline	-0.007	No, it will not be cold, so you won't need a jacket. It will be mostly sunny with a high of 66 and a low of 47 next ~~Wednesday~~
Self-training	-0.001	No, you won't need a jacket, it's not supposed to be cold next ~~Wednesday~~. It'll be mostly sunny with a high of 66 and a low of 47
Full	1.000	No, you don't need a jacket, because it won't be cold next Wednesday. Expect a high of 66 and a low of 47, with mostly sunny skies
User query	-	What will the temperature be tomorrow morning
Template	-	There will be temperatures between 76 and 80 tomorrow morning there will be partly cloudy tomorrow morning
Reference	-	The temperature for tomorrow morning will be between 76 and 80 fahrenheit along with partly cloudy skies
Predictions		
Baseline	-0.002	Expect partly cloudy skies and tomorrow morning. Expect partly cloudy skies and tomorrow morning
Self-training	0.997	Expect it to be between 76 and 80 and partly cloudy tomorrow morning
Full	0.995	The weather for tomorrow morning will be between 76 and 80 with partly cloudy skies

Table 7: A few examples of sample responses from different models on unseen test split for FewShotWeather dataset. We use 1shot-250 train split to fine-tune baseline and BLEURT based self-training. In general, we see that the baseline model generate responses that are repetitive in nature and either contain novel content or are missing relevant details.

Fields	BLEURT	Input or output response
User query	-	On the 12th of this month would be great.
Template	-	Would you like to fly with American Airlines? The onward flight takes off at 4 am. It has a layover. The returning flight takes off at 12:45 pm. The ticket costs $552
Reference	-	How about a connecting American Airlines flight taking off at 4 am and costing $552? The return time is at 12:45 pm.
Predictions		
Baseline	-0.004	Okay. I've found an American Airlines flight departing at 4 am and returning at 12:45 pm. I inform you that the flight has a return flight at 12:45 pm. The ticket is $1052.
Self-training	0.996	I've found an American Airlines flight departing at 4 am and returning at 12:45 pm. This will cost you $552.
Full	0.998	There is an American Airlines flight that leaves at 4 am and has a layover and a return flight at 12:45 pm for $552.
User query	-	Find me the contact number and the cost per night.
Template	-	It costs $93 per night. Their phone number is +44 20 7402 0281.
Reference	-	You can contact them on +44 20 7402 0281 and the cost per night is $93.
Predictions		
Baseline	-0.003	The phone number for Aslam's Rasoi in San Francisco is +44 20 7402 0181.
Self-training	0.998	The cost is $93 per day. The number is +44 20 7402 0281.
Full	0.998	It's $93 per night and you can call them at +44 20 7402 0281.
User query	-	That sounds good. How much is the rental per day?
Template	-	Your car has been reserved. It costs $32.00 per day.
Reference	-	It will cost you $32.00 per day. Your car has been reserved.
Predictions		
Baseline	-0.004	Your car has been reserved. It costs $32,32.00 per day.
Self-training	-0.002	Your car has been reserved. It costs $32.50 per day.
Full	0.998	Your car has been reserved. The cost is $32.00 per day.
User query	-	Yes, set up an appointment for me.
Template	-	What date would you like the appointment for? What is your preferred time for the appointment?
Reference	-	What date and time do you prefer?
Predictions		
Baseline	-0.003	What date ~~and time~~ would you like the appointment?
Self-training	0.804	What date would you like the appointment and what time do you want the appointment?
Full	0.700	What date and time would you like the appointment for?
User query	-	I want to find a property in Palo Alto.
Template	-	Do you want to find a place to rent or buy?
Reference	-	Are you looking to rent or buy?
Predictions		
Baseline	-0.004	Do you want a house ~~for rent or purchase~~?
Self-training	0.999	Do you want a place for rent or purchase?
Full	0.815	Do you want to rent ~~or purchase~~ a place?
User query	-	I also need to find a restaurant there so that I can eat something on that day. The restaurant I am interested in should be considered as reasonably priced, and that is serves Dumplings.
Template	-	I found 1 restaurants you may like. Hong Kong East Ocean Seafood Restaurant is a nice restaurant. It's located in Emeryville.
Reference	-	There is 1 restaurant which matches what you want. A great restaurant found in Emeryville is the Hong Kong East Ocean Seafood Restaurant.
Predictions		
Baseline	-0.003	I have 1 option an it is Hong Kong East Ocean Seafood ~~Cafe located in Emeryville~~.
Self-training	0.857	I have 1 option an it is Hong Kong East Ocean Seafood Cafe located in Emeryville.
Full	0.998	I found 1 restaurant you might like. How about the Hong Kong East Ocean Seafood Restaurant in Emeryville?

Table 8: A few examples of sample responses from different models on unseen test split for FewShotSGD dataset. We use 5-shot train split to fine-tune baseline and BLEURT based self-training. In general, we see that the baseline model fails to generate responses that are coherent and factually grounded in the input.