Structure-to-Text Generation with Self-Training, Acceptability Classifiers and Context-Conditioning for the GEM Shared Task

Shreyan Bakshi,* **Soumya Batra,*** **Peyman Heidari,**
Ankit Arun, Shashank Jain, Michael White*†
Facebook
{shreyanb, sbatra, peymanheidari,
ankitarun, shajain, mwhite14850}@fb.com

Abstract

We explore the use of self-training and acceptability classifiers with pre-trained models for natural language generation in structure-to-text settings using three GEM datasets (E2E, WebNLG-en, Schema-Guided Dialog). With the Schema-Guided Dialog dataset, we also experiment with including multiple turns of context in the input. We find that self-training with reconstruction matching along with acceptability classifier filtering can improve semantic correctness, though gains are limited in the full-data setting. With context-conditioning, we find that including multiple turns in the context encourages the model to align with the user's word and phrasing choices as well as to generate more self-consistent responses. In future versions of the GEM challenge, we encourage the inclusion of few-shot tracks to encourage research on data efficiency.

1 Introduction

Natural Language Generation (NLG) plays a crucial role in task-oriented dialog systems, which have become increasingly commonplace in voice-controlled assistants, customer service agents, and similar systems. In the research community, generative models (Wen et al., 2015; Dušek and Jurcıcek, 2016; Rao et al., 2019) have become popular for their data-driven scaling story and superior naturalness over typical template-based systems (Gatt and Krahmer, 2018; Dale, 2020). However, training reliable and low-latency generative models has typically required tens of thousands of training samples (Balakrishnan et al., 2019; Novikova et al., 2017). From a practical perspective, model maintenance with such a large dataset has proven to be challenging, as it is resource-intensive to debug and fix responses, make stylistic changes, and add new capabilities. As such, it is of paramount importance to investigate ways of bringing up new domains and languages with as few examples as possible while maintaining quality.

Pre-trained models like GPT2 (Radford et al., 2019) have shown great potential to address this challenge (Peng et al., 2020; Chen et al., 2020), and combining pre-trained models with self-training has been shown to improve data efficiency even further (Arun et al., 2020). Additionally, semantic fidelity classifiers (Harkous et al., 2020) can be helpful in addressing issues with semantic correctness that are exacerbated in low-data settings (Anonymous, 2021). Indeed, Heidari et al. (2021) have recently shown that using pre-trained models together with self-training and acceptability classifiers — i.e., classifiers to predict semantic correctness and grammaticality — can play a crucial role in developing a production-quality model with just a few hundred training samples.

In this paper, we apply these techniques to 3 of the datasets from the GEM Shared Task (Gehrmann et al., 2021): the Schema-Guided Dialog (SGD) dataset (Rastogi et al., 2019), the End-to-End (E2E) dataset (Novikova et al., 2017) and the WebNLG-en dataset (Gardent et al., 2017). We focus on these 3 datasets specifically because they mostly closely resemble natural language generation (NLG) in a task-oriented dialog setting, as in Heidari et al.'s work. Although we did not expect substantial gains using these methods in high-data settings, we wanted to try them out on additional datasets in order to better understand their behavior, as well as to encourage research in low-data settings for future editions of the GEM shared task.

With the SGD dataset, we were also particularly interested in the effect of including multiple turns of dialog context in the input, and how this effects the behavior of our NLG system. In early work, Brockmann et al. (2005) showed that cache-based language models can be used to adapt NLG systems

*Equal Contribution
†Work done while on leave from Ohio State University

Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021), pages 136–147

to align with user's language, while subsequent work investigated structural priming more specifically (Reitter et al., 2006) and the impact of such adaptation in deployed dialog systems (Stoyanchev and Stent, 2009). Dušek and Jurčíček (2016) investigated ways of adapting to the user's way of speaking with neural models using the previous user turn; more recently, Kale and Rastogi (2020) demonstrated with the SGD dataset that including multiple turns of context in the input to a pretrained model yields large gains in BLEU scores. However, Kale & Rastogi did not analyze the reasons underlying these gains; here we show that context-conditioning does indeed enable the model to better align with the user's word and phrasing choices, though self-consistency with previous system turns is an even stronger factor.

2 Methods

2.1 Context-Conditioning and Templatizing Inputs

For the Schema-Guided Dialog Dataset, we included the service in the input (Table 1) after our initial experiments indicated that the service was crucial to generating accurate responses for some dialog acts (e.g., `Notify_Failure`). We notified the organizers of this issue, and they released an enhanced version of the dataset including this information. We also experimented with sorting the inputs and conditioning on 1–5 turns of context.

Following Kale and Rastogi (2020), we also tried converting the inputs into semi-natural text (Table 2) using their templates. These templates aim to provide minimal coverage of the input dialog acts rather than actually producing natural outputs, as that task is left to the pre-trained model to learn (for that reason, we call them *templatized* inputs rather than template-based inputs).

To use the Kale & Rastogi templates, we found that it was additionally necessary to augment the dialog acts with the service call method in some cases. Consequently, we retrieved this information from the original Schema-Guided Dialog dataset, sharing a script for doing so with the organizers.

2.2 Tree-Structured Ordering

For the WebNLG dataset, we followed Yang et al. (2020) in ordering the input triples using their implicit tree structure. Yang et al. found that traversing the tree in depth-first search order yielded substantial improvements in their experiments that were competitive with using a learned input ordering. Given the tendency to put heavier constituents towards the end of a sentence in English (Hawkins, 1994; Gibson, 2000; Temperley, 2007; Rajkumar et al., 2016), we additionally sorted siblings by increasing subtree depth, breaking ties by sorting alphabetically on predicate names.

To format the input data, we followed Li et al. (2020) in separating subjects, predicates and objects with separators while replacing underscores with spaces and removing quotes; we also prepended the category with a separator. An example input appears in Table 3.

Algorithm 1: Self-Training via Reconstruction

Start with labeled data $\mathcal{L}$ and unlabeled data $\mathcal{U}$, with inputs $\mathcal{X}$ and outputs/labels $\mathcal{Y}$;

Set current pseudo-labeled data $\mathcal{L}' := \mathcal{L}$;

repeat

  Train 2 models on $\mathcal{L}'$ (in parallel):
    Generation model $\mathcal{G}$ from $\mathcal{X} \rightarrow \mathcal{Y}$;
    Recon. model $\mathcal{R}$ from $\mathcal{Y} \rightarrow \mathcal{X}$;

  Run $\mathcal{G}$ on $\mathcal{U}$ to get pseudo-labels $\mathcal{Y}'$;
  Run $\mathcal{R}$ on $\mathcal{Y}'$ to get recon. inputs $\mathcal{X}'$;

  $\mathcal{L}' := \mathcal{L} \cup \{\text{rows where } X = X'\}$;

until *convergence or maximum iteration*;

2.3 Self-Training

Annotating large quantities of high-quality data is time and resource consuming. However, it is often possible to automatically generate a lot of unlabeled data using a synthetic framework. Semi-supervised techniques can then be applied based on this mix of labeled and unlabeled data, to improve model performance.

Since the datasets do not come with unpaired inputs, we create such inputs for self-training via automatic deletion of all combinations of parts of the (structured) input query, to generate larger sets of unlabeled data for self-training. For each original input, we randomly select up to 20 unpaired inputs created via deletion. Note that with WebNLG, deletion is constrained to yield connected subtrees.

Unsorted	Buses_2 __**sep**__ OFFER departure_time 8:30 am, OFFER **price** $23, OFFER **fare_type** Economy
Sorted	Buses_2 __**sep**__ OFFER **departure_time** 8:30 am, OFFER **fare_type** Economy, OFFER **price** $23
Prompt Sorted	Buses_2 __**sep**__ OFFER **departure_time** 8:30 am, OFFER **fare_type** Economy, OFFER **price** $23 __**sep**__ **user:** Do you have any other buses available?
Context 5 Sorted	Buses_2 __**sep**__ OFFER **departure_time** 8:30 am, OFFER **fare_type** Economy, OFFER **price** $23 __**sep**__ **user:** I am traveling from Sacramento, CA to SFO on March 7th. **sys:** I have found a bus that departs at 7:40 am. The economy ticket is priced at $22 **user:** What are the stations of arrival and departure? **sys:** It starts from Sacramento Valley Station and arrives at Salesforce Transit Center. **user:** Do you have any other buses available?

Table 1: Context-Conditioned and Sorted Inputs for the SGD Dataset (with the service name)

Template	Buses_2 __**sep**__ **How about a bus leaving at 8:30 am and the price of the ticket is $23.** **It is Economy ticket.**
Template Prompt	Buses_2 __**sep**__ How about a bus leaving at 8:30 am and the price of the ticket is $23. It is Economy ticket. __**sep**__ **user:** Do you have any other buses available?
Template Context 5	Buses_2 __**sep**__ How about a bus leaving at 8:30 am and the price of the ticket is $23. It is Economy ticket. __**sep**__ **user:** I am traveling from Sacramento, CA to SFO on March 7th. **sys:** I have found a bus that departs at 7:40 am. The economy ticket is priced at $22 **user:** What are the stations of arrival and departure? **sys:** It starts from Sacramento Valley Station and arrives at Salesforce Transit Center. **user:** Do you have any other buses available?

Table 2: Templatized and Context-Conditioned Inputs for the SGD Dataset

Original	Politician, [Poland | language | Polish_language, Adam_Koc | nationality | Poland, Poland | ethnicGroup | Kashubians]
Tree-Structured (DFS)	Politician __**sep**__ __subj__ Adam Koc __pred__ nationality __obj__ Poland __subj__ Poland __pred__ ethnic group __obj__ Kashubians __subj__ Poland __pred__ language __obj__ Polish language

Table 3: Tree-Structured Ordering Inputs for the WebNLG Dataset

Most approaches to self-training for NLG—including earlier work on automatic data cleaning—make use of cycle consistency between parsing and generation models (Chisholm et al., 2017; Nie et al., 2019; Kedzie and McKeown, 2019; Qader et al., 2019). More recently, Chang et al. (2021) have developed a method for randomly generating new text samples with GPT-2 then automatically pairing them with data samples. Our approach, following Heidari et al. (2021), likewise takes advantage of pre-trained models; by comparison though, we take a much more direct approach to generating new text samples from unpaired inputs in self-training. As described formally in Algorithm 1, self-training here consists of multiple cycles of generation and reconstruction. Note that unlike work in MT that employs back-translation, including unsupervised MT (Lample et al., 2018), we do not assume access to large amounts of target text. Additionally, unlike He et al.'s (2020) self-training approach to MT, we make use of reconstruction matching to filter the pseudo-annotated data (line 14) in each self-training iteration.[1]

We fine-tune BART (Lewis et al., 2020), a pre-trained seq2seq language model, for both steps. For generation, we train a BART large model to produce the responses given the scenario. In parallel, the same generation data is used to fine-tune a reconstruction BART large model to obtain the generation input, given the responses. After generation in each cycle, we use the reconstruction model to select samples with exact reconstruction match. Finally, the selected samples are added to the training pool for the next self-training cycle.

We noted that for the case of SGD, the self-trained model was susceptible to stuttering, i.e., repeating the same phrase over and over again (this occurred in $< 1\%$ of the validation samples). This was not observed in the BART-Large generation model. Hence, to control for stuttering, for each response generated by the self-trained model, we used the heuristic that if any word (excluding stop words such as articles, conjunctions, etc.) was repeated in the generated response more than 5 times, we substituted the response generated by the BART-Large model instead.

[1]He et al. find it useful to fine-tune the model on just the labeled data at the end of each iteration; we leave experimenting with this additional step in our setting to future work.

2.4 Filtering via Acceptability Classifiers

Based on work by (Anonymous, 2021), we trained acceptability classifiers for each dataset using the training data available for its generation model. A response is considered (minimally) acceptable if it is both semantically accurate and grammatical.

As per Anonymous (2021)'s recommendation, since we don't have any representative validation set of labelled acceptable/unacceptable samples, we took a BART-Large model and finetuned it on the training set. Next, we used MaskFilling strategy to generate synthetic acceptable/unacceptable samples wherein we inserted 3 to 7 random masks to the seed data (i.e. training data for generation model) and used the fine-tuned BART model to fill in the masks. This helped capture similar patterns in the seed data and masked words in the response are replaced by tokens most similar to that in seed data, thereby generating more realistic unacceptable samples.

We then passed each of the generated synthetic samples to a RoBERTa-based entailment model and partitioned samples that had a 2-way entailment with respect to the original seed sample as acceptable and the rest unacceptable. In addition, we ensured that that the BLEU score between synthetic sample and original seed sample was between 0.5-0.9 for unacceptable class and above 0.9 for acceptable class. Since the BART masking method will only generate paraphrases with similar sentence structure due to masks insertion in the original seed responses thereby maintaining the original sub-sequences order, these paraphrases tend to differ only slightly compared to the original responses. Hence, a BLEU score $>$0.9 allows us to capture most of them while a BLEU score $>$0.5 ensures that we are only selecting unacceptable samples with nuanced errors.

Finally, we trained a RoBERTa-base classifier over the acceptable and unacceptable classes. At inference time, we passed the n-best responses obtained by the self-trained generation model through the trained acceptability classifier. We filtered out the responses that had a high unacceptability score (threshold determined over validation set for each dataset). Of the remaining responses, we selected the top response. In case all responses were filtered out, we selected the top response from the original n-best list.

	BART Base			BART Large		
	Unsorted	Sorted	Template	Unsorted	Sorted	Template
No Context	34.39	34.78	35.99	35.01	35.09	36.48
Prompt	37.72	37.90	39.03	38.96	39.01	39.99
Context 5	43.37	43.55	**44.18**	44.75	43.79	**45.21**

Table 4: BLEU scores for Schema-Guided Dialog validation set

	E2E Self-Train			WebNLG-en Self-Train			SGD Self-Train		
	Initial	Round 1	Round 2	Initial	Round 1	Round 2	Initial	Round 1	Round 2
500 Rows	67.85	81.58	83.83	54.41	65.87	52.55	53.87	61.68	62.91
10% Data	86.04	85.72	86.51	76.90	80.44	82.78	63.25	64.14	63.98
Full Data	89.18	90.46	**91.77**	85.24	85.78	**85.90**	63.95	63.78	**64.23**

Table 5: Exact Reconstruction Match % on full validation set for End-to-End, WebNLG-en and Schema-Guided Dialog datasets when self-trained starting with varying amounts of seed data

3 Results

3.1 Context-Conditioning and Templatizing Inputs

The BLEU (Papineni et al., 2002) scores for various BART models on the Schema-Guided Dialog validation set appear in Table 4.[2] As the Table shows, sorting the standard inputs appears to yield a small improvement. Templatizing the inputs yields a larger gain, over 1 BLEU point in some cases. Using BART Large yields a somewhat smaller gain over using BART Base, but the gains are around another BLEU point when used with templatized inputs and context. By comparison, using the dialog context yields very large gains, with including the prompt in the input adding over 3 BLEU points, and adding another four turns of context to the input improving another 5 BLEU points or so. These gains corroborate the ones reported by Kale and Rastogi (2020) using T5 (Raffel et al., 2020), while also putting them in the context of improvements based on model size and type of input. We plan to make our additional baseline results above publicly available in the near future.

3.2 Self-Training

We ran self-training as described in Algorithm 1 on all 3 datasets, with multiple variations for each including few-shot, low data and full data settings. The BLEU scores with self-training do not improve significantly over the regular training paradigm. However, we observe sharp increase in the exact reconstruction match rate on the validation set when using self-training, especially in the lower data regimes, as shown in Table 5. This metric is calculated by training a reconstruction model on the full labeled data once in the beginning, and then using this model to perform reconstructions at different stages during self-training – observing its performance on 100% of the validation set each time, for automatic evaluation purposes. Note that with the SGD dataset, we used reconstruction accuracy on the sorted input for this evaluation, as we observed some issues with reconstructing the textualized input; these are discussed further in the next section.

[2]These BLEU scores are calculated with a different version of BLEU than used by the GEM metrics; the BLEU score for the best model according to the GEM metrics is 43.35.

3.3 Filtering via Acceptability Classifiers

We ran n-best response filtering using Acceptability Classifiers on the outputs of the BART-Large generation model as described in 2.4. The BLEU scores and reconstruction exact match rate only slightly changed (increased or decreased) at different unacceptability confidence thresholds.

We also ran a RoBERTa-based entailment model on the small number of responses that were changed by the acceptability classifier with respect to the target reference, as well as on the corresponding 1-best response from the generation model. We estimated number of paraphrases by checking for 2-way entailment between the pairs. We observed a slight increase in the total number of paraphrases identified using this model when filtering via Acceptability Classifier, as shown in Table 6. Examples of positive changes appear in Table 7.

	Total number of paraphrases wrt target reference		
	Total Changed	Response chosen by acc	1-best Response
WebNLG	112	105	104
E2E	100	68	64
SGD	22	12	15

Table 6: Number of paraphrases identified by RoBERTa-base entailment model when response chosen by Acceptability Classifier (acc) filtering method (at best threshold) compared to the 1-best response from vanilla BART-Large generation method on validation sets.

Dataset	Input	Response chosen by acc	1-best Response
WebNLG	Food __sep__ __subj__ Arem-arem __pred__ country __obj__ Indonesia __subj__ Indonesia __pred__ leader __obj__ Joko Widodo __subj__ Indonesia __pred__ leader __obj__ Jusuf Kalla	Arem arem **originates** from Indonesia where Joko Widodo and Jusuf Kalla are leaders.	Joko Widodo and Jusuf Kalla are leaders in Indonesia where Arem-arem is a **traditional** dish.
E2E	name[The Wrestlers], customer rating[5 out of 5], familyFriendly[yes]	The Wrestlers is a 5 out of 5 rated family friendly **venue**.	The Wrestlers is a five star, family friendly **sushi bar**.
SGD	Services_4 __sep__ REQUEST type Psychologist Psychiatrist	Do you need a Psychiatrist or a **Psychologist**?	Do you need a Psychiatrist or a **Psychiatrist**?

Table 7: Sample Responses chosen by Acceptability Classifier (acc) filtering over 1-best response

3.4 Combined Methods

Results from the GEM metrics on the validation set when using the Acceptability Classifier with the self-trained BART-Large models appear in Table 8.[3]

4 Analysis

4.1 Context-Conditioning and Templatizing Inputs

Here we analyze the effects of including multiple turns of context in the input. Table 9 shows examples of how the model that takes five previous turns of context as input (Context 5) aligns with aspects of the context more strongly than the model that takes just one turn of context as input (Prompt). Examples (a) and (b) show how the Context 5 models generates wordier or more concise outputs depending on the user's previous word and phrase choices, while Example (c) shows how the Context 5 model instead picks up on its own previous phrasings to yield a more consistent way presenting similar weather information across responses.

These effects can be verified quantitatively as well. Table 10 shows how the Context 5 model's responses correlate more strongly in length with both previous user and system turns, and Table 11 similarly shows that BLEU-2 scores against the context are more similar for the Context 5 model than the Prompt model. Finally, Table 12 shows that these contextual BLEU-2 scores are positively correlated with BLEU scores against the reference. (All correlations are statistically significant, albeit weak.)

4.2 Self-Training

Since we did not observe an increase in BLEU scores with self-training in the full-data setting, we manually examined a sample of validation set outputs for the initial, supervised BART-Large model in comparison to the self-trained BART-Large model where these outputs differed in reconstruction accuracy. Across all 3 datasets, we found that both outputs were usually good, reflecting issues with the reconstruction model or our way of determining a reconstruction match, rather than real differences in the semantic correctness of the outputs. However, in the cases where real semantic differences were found, we observed that the changes were generally in the direction of improved semantic correctness with the self-trained model.

In calculating reconstruction accuracy, we noticed many issues that can be considered cases of inadequate normalization. For example, with the E2E dataset, the customer rating and price range slots use mostly interchangeable values in the input such as "5 out of 5" and "high" as values for top-rated venues; this means that the reconstruction

[3]Note that METEOR scores here are computed via NLTK

	BLEU	METEOR	ROUGE-L	BERTScore	BLEURT
E2E	34.54	0.578	54.5	0.916	0.292
WebNLG-en	68.74	0.777	72.1	0.959	0.478
SGD	43.38	0.560	60.8	0.898	0.177

Table 8: Validation set automatic metrics for self-trained models with Acceptability Classifier filtering

	Content	Context	Reference	Prompt	Context 5
(a)	How many tickets would you like?	*user*: Okay. **Can you** find **me** a hotel in that area, so that I will have a place to stay in? *sys*: ... *user*: **Can you** give **me** their phone number? How much will it cost me per nigh? *sys*: ... *user*: That is nice. Now I want to buy tickets for the event you found earlier.	**Can you** tell **me** the number of tickets you want to buy?	How many tickets do you want to buy?	**Can you** tell **me** the number of tickets you want to buy?
(b)	Your reservation is successful. They do not have outdoor seating.	*user*: **Sounds good** to me. *sys*: ... *user*: Sure, **book it** for 11:00 *sys*: ... *user*: **Perfect**. do they have outdoor seating?	**Booking confirmed**. They **don't** have outdoor seating.	Your reservation has been made. They do not have outdoor seating.	**Booking confirmed**. They **don't** have outdoor seating.
(c)	The average temperature for the day should be 87 degrees Fahrenheit. There is a 3 percent chance of rain.	*user*: Duncans Mills *sys*: **It will be** 93 **degrees with a** 20 **percent chance of rain**. *user*: How about on the 5th of this month? *sys*: **It will be about** 90 **degrees with a** 1 **percent chance of rain**. *user*: How about in Mexico city?	**It will be** 87 **degrees with a** 3 **percent chance of rain**.	Average temperature: 87 degrees Fahrenheit. Chance of rain: 3 percent.	**It will be about** 87 **degrees with a** 3 **percent chance of rain**.

Table 9: Examples illustrating model adaptation to the dialog context when using 5 previous turns of context (Context 5) vs. just one previous turn (Prompt). Example (a) shows how the Context 5 model picks up on the user's wordier phrasing, leading to an exact match with the reference. Example (b) indicates how the Context 5 model instead uses a more concise phrasing, picking up on the user's terseness. Example (c) shows how the Context 5 model instead picks up on its own previous phrasings to yield a self-consistent way of presenting similar weather information for different locales and dates.

	User	System
Reference	0.337	0.095
Prompt	0.275	0.025
Context 5	0.320	0.085

Table 10: Correlations in model turn length using 5 previous turns of context (Context 5) vs. just one previous turn (Prompt) with user and system turns in the preceding context (5 turns), in comparison to reference.

	User	System
Reference	15.24	15.68
Prompt	8.80	13.11
Context 5	15.88	17.29

Table 11: Mean model BLEU-2 scores (with no length penalty) using 5 previous turns of context (Context 5) vs. just one previous turn (Prompt) against user and system turns in the preceding context (5 turns), in comparison to reference.

model essentially has to guess which one actually appeared in the input. In future work, we intend to add compare the set of slots with normalized values rather than just using exact string match. Similar issues arose with WebNLG, where the reconstruction model had difficulty getting the order of the triples correct, and with SGD, where we discovered that similar but non-identical templates across related services caused confusion for the reconstruction model. Additionally, with SGD we observed that making the dialog context available as input to the

	User	System
Prompt	0.088	0.131
Context 5	0.083	0.204

Table 12: Correlations between contextual BLEU-2 scores (with no length penalty) for model using 5 previous turns of context (Context 5) vs. just one previous turn (Prompt) against user and system turns with BLEU scores (against reference).

reconstruction model would be helpful in many cases, since many responses employing elliptical constructions were difficult for the reconstruction model (despite being clear and natural in context).

4.3 Acceptability Classifier Filtering

Looking more closely at a random sample of the responses that were changed by the acceptability classifier, we noted that the acceptability classifier filtering indeed usually chooses a better response than the default in high confidence unacceptability regions. This also makes intuitive sense as we expect the generation model to be correct and fluent most of the time and acceptability classifier filtering helping in a small number of cases. We expect this impact to be higher on cases which are not represented in the training distribution.

5 Discussion

It is fascinating that simply including multiple turns of preceding dialog in the input to a pre-trained model has such a large impact on generated responses, and in particular that doing so increases alignment with the user's language as well as consistency with the system's own previous responses. Both factors can be expected to enhance naturalness, though this will need verification via human evaluation. More compellingly, it is likely that these effects will enhance user perceptions of the system in an extrinsic evaluation of how NLG affects perceived dialog quality. To verify such effects, it will be important to study context-enhanced NLG in the context of actual dialogs with users, rather than in a simpler overhearer paradigm.

Turning to self-training, it is clear from our experiments that gains in semantic correctness can be quite large in low-data settings. Moreover, the pay-off from acceptability classifier filtering can be expected to be larger there. Nevertheless, gains in low-data settings have generally not brought systems fully in line with those trained in high-data settings. As such, there remains considerable room for improvement in such low-data settings, even when using pre-trained models. To promote work along these lines, future editions of the GEM shared task could have few-shot tracks where the number of samples for supervised training is quite limited. Moreover, it would be extremely helpful to make unpaired inputs available for these tracks. While creating unpaired inputs via deletion is somewhat helpful, this technique cannot help with unseen or few-shot test items in the final test set. As such, providing unpaired inputs corresponding to these few-shot test items would provide a way to experiment in a standardized fashion with methods for generalizing in these cases. Note that in the case of datasets created via simulation, as with the SGD dataset and its dialog simulator, creating new unpaired inputs would only require running the simulator for the few-shot domains. Doing so for a shared task should be much easier than releasing all the code used during dataset creation, so we urge the organizers to consider this option in future.

Acknowledgments

We thank the anonymous reviewers for their helpful comments. We also thank Arash Einolghozati, Lee Callender, Catharine Youngs, Anuj Kumar, Shawn Mei, Sonal Gupta, Pinar Donmez and Vikas Bhardwaj for helpful discussion.

References

Anonymous. 2021. Building adaptive acceptability classifiers for neural NLG. Under review.

Ankit Arun, Soumya Batra, Vikas Bhardwaj, Ashwini Challa, Pinar Donmez, Peyman Heidari, Hakan Inan, Shashank Jain, Anuj Kumar, Shawn Mei, Karthik Mohan, and Michael White. 2020. Best practices for data-efficient modeling in NLG:how to train production-ready neural models with less data. In *Proceedings of the 28th International Conference on Computational Linguistics: Industry Track*, pages 64–77, Online. International Committee on Computational Linguistics.

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*.

Carsten Brockmann, Amy Isard, Jon Oberlander, and Michael White. 2005. Modelling alignment for affective dialogue. In *Proc. of the Workshop on Adapting the Interaction Style to Affective Factors at the 10th International Conference on User Modeling (UM-05)*.

Ernie Chang, Xiaoyu Shen, Dawei Zhu, Vera Demberg, and Hui Su. 2021. Neural data-to-text generation with LM-based text augmentation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 758–768, Online. Association for Computational Linguistics.

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.

Andrew Chisholm, Will Radford, and Ben Hachey. 2017. Learning to generate one-sentence biographies from Wikidata. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 633–642, Valencia, Spain. Association for Computational Linguistics.

Robert Dale. 2020. Natural language generation: The commercial state of the art in 2020. *Natural Language Engineering*. To appear.

Ondřej Dušek and Filip Jurčíček. 2016. A context-aware natural language generator for dialogue systems. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 185–190, Los Angeles. Association for Computational Linguistics.

Ondrej Dušek and Filip Jurcıcek. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. In *The 54th Annual Meeting of the Association for Computational Linguistics*, page 45.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for NLG micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 179–188. Association for Computational Linguistics.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

Sebastian Gehrmann, Tosin P. Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Aremu Anuoluwapo, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh D. Dhole, Wanyu Du, Esin Durmus, Ondrej Dusek, Chris Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Rubungo Andre Niyongabo, Salomey Osei, Ankur P. Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. The GEM benchmark: Natural language generation, its evaluation and metrics. *CoRR*, abs/2102.01672.

Edward Gibson. 2000. The dependency locality theory: A distance-based theory of linguistic complexity. *Image, language, brain*, 2000:95–126.

Hamza Harkous, Isabel Groves, and Amir Saffari. 2020. Have your text and use it too! end-to-end neural data-to-text generation with semantic fidelity. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2410–2424, Barcelona, Spain (Online). International Committee on Computational Linguistics.

John A Hawkins. 1994. *A performance theory of order and constituency*. 73. Cambridge University Press.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2020. Revisiting self-training for neural sequence generation.

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 SIGDIAL*. To appear.

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.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049, Brussels, Belgium. 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.

Xintong Li, Aleksandre Maskharashvili, Symon Jory Stevens-Guille, and Michael White. 2020. Leveraging large pretrained models for WebNLG 2020. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 117–124, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Feng Nie, Jin-Ge Yao, Jinpeng Wang, Rong Pan, and Chin-Yew Lin. 2019. A simple recipe towards reducing hallucination in neural surface realisation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2673–2679, Florence, Italy. Association for Computational Linguistics.

Jekaterina Novikova, Ondrej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Saarbrücken, Germany. ArXiv:1706.09254.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. *In Proc. ACL-02*.

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. *Empirical Methods in Natural Language Processing (EMNLP)*.

Raheel Qader, François Portet, and Cyril Labbé. 2019. Semi-supervised neural text generation by joint learning of natural language generation and natural language understanding models. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 552–562, Tokyo, Japan. Association for Computational Linguistics.

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

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.

Rajakrishnan Rajkumar, Marten van Schijndel, Michael White, and William Schuler. 2016. Investigating locality effects and surprisal in written english syntactic choice phenomena. *Cognition*, 155:204–232.

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. 2019. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *arXiv preprint arXiv:1909.05855*.

David Reitter, Frank Keller, and Johanna D. Moore. 2006. Computational modelling of structural priming in dialogue. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 121–124, New York City, USA. Association for Computational Linguistics.

Svetlana Stoyanchev and Amanda Stent. 2009. Lexical and syntactic adaptation and their impact in deployed spoken dialog systems. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 189–192, Boulder, Colorado. Association for Computational Linguistics.

D. Temperley. 2007. Minimization of dependency length in written english. *Cognition*, 105:300–333.

Tsung-Hsien Wen, Milica Gasic, 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. Association for Computational Linguistics.

Zixiaofan Yang, Arash Einolghozati, Hakan Inan, Keith Diedrick, Angela Fan, Pinar Donmez, and Sonal Gupta. 2020. Improving text-to-text pre-trained models for the graph-to-text task. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 107–116, Dublin, Ireland (Virtual). Association for Computational Linguistics.

A Appendix

A.1 Model Hyperparameters

Model hyperparameters appear in Tables 13–15. In addition, the best performing model on the validation set had the unacceptability confidence thresholds for filtering listed in Table 16. The bounds used to calculate the thresholds were [0.1–0.9] with 0.1 step size.

A.2 Computing Infrastructure

For training each generation BART-Large model, 8 GPUs were used, which took about 3.5 hours for larger datasets like SGD.

For training the accuracy classifier RoBERTa-base model, 8 GPUs were also used, taking up to 2 days on larger datasets like SGD including data preparation and model training time.

All experiments were conducted on 32GB Quadro GV100 GPUs. The GPUs are part of a shared distributed cluster, which adds its own time overheads.

Tokenizer	BPE
Tokenizer Max Length	256
Dropout	0.3
Encoder/Decoder Embedding Dim	1024
Optimizer	Adam
LR	0.000005
Weight Decay	0.00001
# Model Params	514484225

Table 13: BART-Large Generation/Reconstruction Hyperparameters

Tokenizer	BPE
Tokenizer Max Length	1024
Encoder output dropout	0.1
Encoder embedding dim	768
# encoder layers	12
# encoder attention heads	12
Decoder dropout	0
Decoder activation	relu
Optimizer	Adam
LR	0.000001
Adam betas	0.9, 0.999
Weight Decay	0
# Model Params	124055810

Table 14: Acceptability Classifier RoBERTa-Base Hyperparameters

Beam Size	5
topk	3
Mask normal	0.5
Mask insert	0.3

Table 15: Acceptability Classifier Data Generation Hyperparameters

Dataset	Unacceptability Threshold
hline E2E	0.6
WebNLG-en	0.7
SGD	0.6

Table 16: Acceptability Classifier Thresholds