# Curriculum-Based Self-Training Makes Better Few-Shot Learners for Data-to-Text Generation

**Pei Ke**[1] , **Haozhe Ji**[1] , **Zhenyu Yang**[2] , **Yi Huang**[3,4] , **Junlan Feng**[3,4] ,
**Xiaoyan Zhu**[1] and **Minlie Huang**[1,4*]

[1]CoAI Group, DCST, IAI, BNRIST, Tsinghua University, Beijing, China

[2]OPPO Mobile Telecommunications Corp., Ltd, China

[3]JIUTIAN Team, China Mobile Research Institute, Beijing 100053, China

[4]Tsinghua University-China Mobile Communications Group Co., Ltd. Joint Institute, Beijing, China

{kp17, jhz20}@mails.tsinghua.edu.cn, yangzhenyu@oppo.com

{huangyi, fengjunlan}@chinamobile.com, {zxy-dcs, aihuang}@tsinghua.edu.cn

## Abstract

Despite the success of text-to-text pre-trained models in various natural language generation (NLG) tasks, the generation performance is largely restricted by the number of labeled data in downstream tasks, particularly in data-to-text generation tasks. Existing works mostly utilize abundant unlabeled structured data to conduct unsupervised pre-training for task adaption, which fail to model the complex relationship between source structured data and target texts. Thus, we introduce self-training as a better few-shot learner than task-adaptive pre-training, which explicitly captures this relationship via pseudo-labeled data generated by the pre-trained model. To alleviate the side-effect of low-quality pseudo-labeled data during self-training, we propose a novel method called Curriculum-Based Self-Training (CBST) to effectively leverage unlabeled data in a rearranged order determined by the difficulty of text generation. Experimental results show that our method can outperform fine-tuning and task-adaptive pre-training methods, and achieve state-of-the-art performance in the few-shot setting of data-to-text generation.

## 1 Introduction

Recently, text-to-text pre-trained models like BART [Lewis *et al.*, 2020] and T5 [Raffel *et al.*, 2020] have emerged in the field of natural language generation (NLG). Their main idea is to capture the relationship among texts by reconstructing original sentences with the input of corrupted sentences. These models can achieve state-of-the-art performance in various NLG tasks via fine-tuning on downstream datasets.

Despite the success of text-to-text pre-trained models, it's challenging to directly apply them to few-shot NLG tasks because their performance can be largely restricted by the number of labeled data [Chen *et al.*, 2020a; Chen *et al.*, 2020b]. Especially in the task of few-shot data-to-text generation, it's
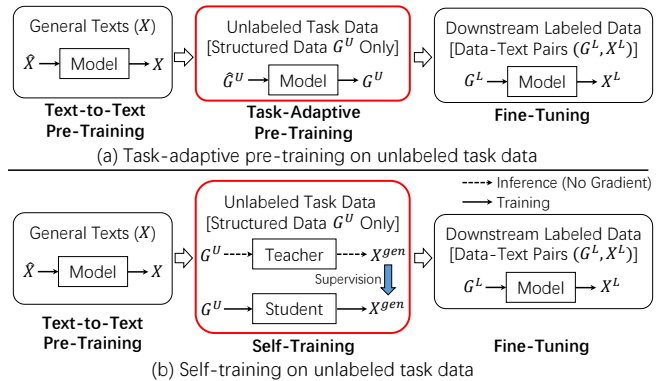
---

*Corresponding author



Figure 1: Comparison between task-adaptive pre-training and self-training in few-shot data-to-text generation. Task-adaptive pre-training adopts unsupervised pre-training methods on the unlabeled task data $G^U$ (such as structured data reconstruction), while self-training instead generates texts $X^{gen}$ for $G^U$ with the teacher model and trains the student model on the generated pseudo-labeled data. $\hat{G}^U$ / $\hat{X}$ denotes the corrupted structured data / texts.

hard to learn the complex relationship between source structured data and target texts via limited labeled data [Gong *et al.*, 2020]. Existing works commonly continue pre-training on large amounts of structured data without corresponding texts for task adaption, aiming to enhance the model performance [Gururangan *et al.*, 2020]. As shown in Figure 1(a), this task-adaptive pre-training strategy appears between general pre-training and fine-tuning, whose purpose is to adapt general pre-trained models (such as BART) to specific tasks (such as data-to-text generation) using unlabeled task data [Xing and Wan, 2021]. We argue that task-adaptive pre-training only captures the relationship among structured data but fails to explicitly model the alignment between structured data and texts, thereby restricting the model performance on the few-shot data-to-text generation tasks.

In this paper, we introduce self-training [Scudder, 1965] as a better few-shot learner for data-to-text generation. As shown in Figure 1(b), self-training is a teacher-student framework where the teacher model creates synthetic labels for

the unlabeled data, and the student model is trained on the pseudo-labeled data constructed by the teacher model. Compared with task-adaptive pre-training, self-training explicitly models the relationship between structured data and texts by training the student model on the pseudo-labeled data generated by the teacher model, instead of solely conducting unsupervised pre-training on the unlabeled structured data. We argue that the main challenge falls into the quality of pseudo-labeled data. Even if the teacher model is initialized with pre-trained models, it may generate low-quality texts especially when dealing with the structured data with complex topologies, which hurt the performance of the student model.

Thus, we present a novel method called Curriculum-Based Self-Training (CBST) to address the challenge. This method utilizes curriculum learning [Bengio *et al.*, 2009] to construct pseudo-labeled data from easy cases to hard ones, and leverages such data into the training process at different iterations. Specifically, we divide the unlabeled dataset into different subsets based on some difficulty metric such as the number of input triples [Ribeiro *et al.*, 2020b]. At each iteration, we first collect the unlabeled data that satisfy the difficulty metric of the current curriculum. Then, we generate synthetic texts for these data using the teacher model, and select the pseudo-labeled data based on the coverage and generation probability. Finally, we train the student model on the labeled data and the selected pseudo-labeled data, and make it act as the teacher model at the next iteration. This method is expected to gradually increase the difficulty of generating texts for unlabeled structured data and improve the quality of pseudo-labeled data constructed by the teacher model.

Our contributions are mainly as follows[1]:

- We introduce self-training to improve text-to-text pre-training for few-shot data-to-text generation. This method explicitly captures the relationship between structured data and texts by generating texts for unlabeled structured data with the teacher model and training the student model on the pseudo-labeled data.

- We propose a novel method called CBST to alleviate the side-effect of low-quality pseudo-labeled data, which introduces curriculum learning to effectively leverage the unlabeled structured data into the training process in the order determined by the difficulty metric.

- We conduct extensive experiments in the few-shot setting of WebNLG and WikiBio datasets. Results show that our method can outperform fine-tuning and task-adaptive pre-training methods, and achieve state-of-the-art performance on these two benchmarks.

## 2 Related Work

### 2.1 Data-to-Text Generation

Early studies on data-to-text generation mainly focus on how to encode the structural information of input data better. Thus, researchers devise complex encoder structures based on sequential neural networks [Trisedya *et al.*, 2018] and graph neural networks [Ribeiro *et al.*, 2020b] to achieve

this goal. Recently, since text-to-text pre-trained models have shown promising performance in various NLG tasks, some works directly fine-tune these pre-trained models including BART [Lewis *et al.*, 2020] and T5 [Raffel *et al.*, 2020] on data-to-text datasets and report impressive results [Ribeiro *et al.*, 2020a; Kale and Rastogi, 2020]. Other works adapt pre-trained models to data-to-text generation tasks by designing specific pre-training tasks such as reconstructing structured data and texts, which further improve the model performance in both supervised and few-shot settings of downstream datasets [Chen *et al.*, 2020a; Ke *et al.*, 2021; Xing and Wan, 2021].

Compared with existing works, we introduce self-training to explicitly capture the relationship between structured data and texts, instead of solely relying on unsupervised task-adaptive pre-training. Our method is expected to utilize unlabeled data more effectively and further improve text-to-text pre-trained models in few-shot data-to-text generation.

### 2.2 Self-Training

Self-training [Scudder, 1965] is a teacher-student framework to leverage unlabeled data through semi-supervised learning. Self-training has been applied to the tasks of text classification [Du *et al.*, 2021] and generation [He *et al.*, 2020]. With the development of pre-trained models, recent works also show that self-training is complementary to pre-training for various classification tasks [Du *et al.*, 2021].

For comparison, our work is the first attempt to utilize self-training to improve text-to-text pre-trained models for few-shot data-to-text generation. Moreover, we consider the difficulty of generating texts for different structured data and introduce curriculum learning [Bengio *et al.*, 2009] to incorporate unlabeled data into the self-training process at different iterations to improve the quality of pseudo-labeled data.

## 3 Method

### 3.1 Task Definition and Model Overview

Given the text-to-text pre-trained model $f_{\theta_{T2T}}$, the downstream labeled dataset $D_L = \left\{(G_i^L, X_i^L)\right\}_{i=1}^{m}$, and the unlabeled dataset $D_U = \{G_i^U\}_{i=1}^{M}$ where $G_i^L$ / $G_i^U$ denotes the structured data and $X_i^L$ indicates the annotated texts ($M \gg m$), our goal is to obtain a data-to-text pre-trained model which can perform well in the few-shot setting of downstream datasets.

The overview of our proposed method is shown in Figure 2. We follow the existing works to linearize the structured data as a sequence of triples containing subjects, predicates, and objects [Ribeiro *et al.*, 2020a; Chen *et al.*, 2020a] as shown in Figure 3. We first initialize the teacher model $f_{\theta_T}$ with the pre-trained model $f_{\theta_{T2T}}$ and train it on the labeled dataset $D_L$. Then, we train the teacher model and the student model iteratively. At each iteration, the teacher model $f_{\theta_T}$ generates synthetic texts for the unlabeled dataset $D_U^{'}$ which contains the data satisfying the difficulty metric of the current curriculum. Then we select the pseudo-labeled dataset $D_{PL}$ based on the coverage and generation probability of the teacher model. The student model $f_{\theta_S}$ which is also initialized with

---

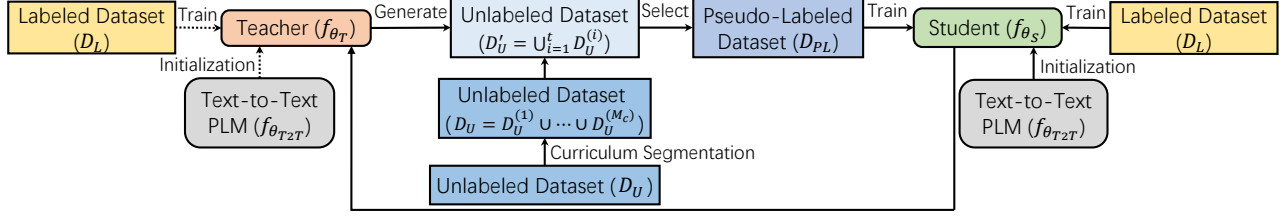[1]The codes are available at https://github.com/kepei1106/CBST.

Figure 2: Overview of CBST. Dotted lines indicate that the teacher is initialized with the text-to-text pre-trained model and trained on the labeled dataset only before the first iteration. Afterward, the teacher model is initialized by the student model from the previous iteration. $M_C$ denotes the number of curriculum, and $D'_U$ contains the unlabeled data included in the current curriculum at each iteration $t$.
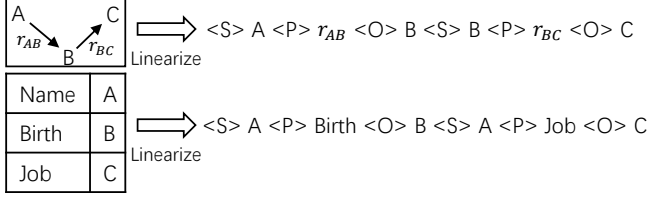


Figure 3: Linearization of structured data including graphs and tables. <S> / <P> / <O> is the special token indicating subjects / predicates / objects, respectively.

the pre-trained model $f_{\theta_{T2T}}$ is trained on the pseudo-labeled dataset $D_{PL}$ and the labeled dataset $D_L$. Finally, the student model $f_{\theta_S}$ acts as the teacher model $f_{\theta_T}$ in the next iteration.

### 3.2 Curriculum-Based Self-Training

Our proposed method is shown in Algorithm 1, which consists of three main steps: curriculum segmentation, pseudo-labeled data construction, and student model training.

**Curriculum Segmentation**

Curriculum segmentation is designed based on the difficulty of generating texts for structured data. In this paper, we choose the number of triples as the difficulty metric to segment the unlabeled dataset $D_U$ into $M_C$ subsets. This metric reflects the complexity of structures in the input data and has a significant impact on the quality of the generated texts [Ribeiro et al., 2020b]. We also try other metrics in §4.6.

**Pseudo-Labeled Data Construction**

This module aims to generate pseudo-labeled data with the teacher model at each iteration. Prior to the first iteration, the teacher model $f_{\theta_T}$ is initialized with the pre-trained model $f_{\theta_{T2T}}$ and trained on the labeled dataset $D_L$ with the following loss function:

$$\mathcal{L}_{init} = \frac{1}{m} \sum_{i=1}^{m} l(X_i^L, G_i^L; f_{\theta_T}) \quad (1)$$

where $l(X, G; f_\theta)$ indicates the negative log-likelihood loss and is commonly used in the sequence generation tasks:

$$l(X, G; f_\theta) = -\sum_{t=1}^{|X|} \log P_\theta(X_t | G, X_{<t}) \quad (2)$$

Note that except for the first iteration, the teacher model is initialized by the student model from the previous iteration.

---

**Algorithm 1** Curriculum-Based Self-Training (CBST)

**Require:**
    Labeled dataset: $D_L = \{(G_i^L, X_i^L)\}_{i=1}^m$
    Unlabeled dataset: $D_U = \{G_i^U\}_{i=1}^M$
    Curriculum segmentation criterion: $C$
    Text-to-text pre-trained model: $f_{\theta_{T2T}}$

**Ensure:**
    Data-to-text pre-trained model: $f_{\theta_{D2T}}$
1: Use $C$ to split $D_U$ into $D_U^{(1)}, D_U^{(2)}, ..., D_U^{(M_C)}$
2: Initialize the teacher model $f_{\theta_T}$ with $f_{\theta_{T2T}}$ and train $f_{\theta_T}$ on $D_L$ via Eq.(1)
3: **for** $t = 1, 2, \cdots, M_C$ **do**
4:     $D'_U \leftarrow \bigcup_{i=1}^t D_U^{(i)}$
5:     Apply $f_{\theta_T}$ to the unlabeled dataset $D'_U$ to acquire the generated texts via Eq.(3)
6:     Select the pseudo-labeled data based on coverage and generation probability to construct the dataset $D_{PL}$
7:     Initialize the student model $f_{\theta_S}$ with $f_{\theta_{T2T}}$ and train $f_{\theta_S}$ on $D_{PL} \cup D_L$ with input noise via Eq.(4)
8:     $f_{\theta_T} \leftarrow f_{\theta_S}$
9: **end for**
10: $f_{\theta_{D2T}} \leftarrow f_{\theta_S}$
11: **return** $f_{\theta_{D2T}}$

---

At each iteration, we first utilize the teacher model to generate the synthetic texts for the unlabeled dataset $D'_U \subset D_U$ which includes the unlabeled structured data that satisfy the difficulty metric of the current curriculum:

$$X_i^{gen} = f_{\theta_T}(G_i^U), i = 1, 2, \cdots, |D'_U| \quad (3)$$

Then, we select the generated texts based on coverage and generation probability which reflect the generation quality [Gehrmann et al., 2018] to construct the pseudo-labeled dataset $D_{PL}$ with $M'$ samples. Specifically, we choose the pseudo-labeled data where the proportion of subjects and objects appearing in the generated text is larger than $\epsilon_{cov}$ and the generation probability of the text ranks top-$\epsilon_{gen}$ to construct the dataset for the student model training. Here, $\epsilon_{cov}$ and $\epsilon_{gen}$ are both hyper-parameters.

**Student Model Training**

The goal of this module is to train the student model on the pseudo-labeled dataset and the labeled dataset. Initialized with the text-to-text pre-trained model $f_{\theta_{T2T}}$, the student

| Word Substitution: Replace words with their synonyms |
| --- |
| Example Input: |
| <S> Bakewell **pudding** <P> **served** <O> Warm or cold |
| Example Output: |
| <S> Bakewell **dessert** <P> **acted** <O> Warm or cold |
| Triple Reordering: Shuffle the order of triples |
| Example Input: |
| <S> Alan Shepard <P> status <O> Deceased |
| <S> Alan Shepard <P> occupation <O> Test pilot |
| Example Output: |
| <S> Alan Shepard <P> occupation <O> Test pilot |
| <S> Alan Shepard <P> status <O> Deceased |

Table 1: Noise functions. <S> / <P> / <O> is the special token indicating the subject / predicate / object of the input triples, respectively. Bold texts indicate the words for substitution.

model $f_{\theta_S}$ is trained with the following loss function:

$$\mathcal{L}_S = \frac{1}{M'} \sum_{i=1}^{M'} l(X_i^{gen}, \text{NF}(G_i^U); f_{\theta_S})$$
$$+ \frac{1}{m} \sum_{i=1}^{m} l(X_i^L, G_i^L; f_{\theta_S}) \qquad (4)$$

where $\text{NF}(G)$ denotes the noise function perturbing the structured data $G$. Existing works show that input noise is beneficial for self-training because it increases the smoothness of the student model [He et al., 2020]. Inspired by the existing works [Jia et al., 2019; Hoyle et al., 2021], we devise two-level noise functions as shown in Table 1, including word substitution at the semantic level and triple reordering at the structural level. Specifically, we substitute each word by its synonym with the probability $p_{word}$ and shuffle the order of triples in each sample with the probability $p_{triple}$, where $p_{word}$ and $p_{triple}$ are hyper-parameters. These functions are expected to perturb the input structured data while largely keeping the semantic and structural information.

In practice, we adopt a separate training strategy which shows better performance than joint training [He et al., 2020]. This strategy first trains the student model on the pseudo-labeled dataset $D_{PL}$ with the first term of $\mathcal{L}_S$, and then trains it on the labeled dataset $D_L$ with the second term.

## 4 Experiment

### 4.1 Dataset

**WebNLG.** This dataset aims to generate textual descriptions for RDF triples [Shimorina and Gardent, 2018]. The number of instances in training / validation / test set is 34,352 / 4,316 / 4,224, respectively. We followed the existing works [Chen et al., 2020a] to pre-process this dataset and use 0.5%, 1%, 5%, 10% of the training instances as the labeled datasets in the few-shot setting.

**WikiBio.** This dataset aims to generate the first sentence of biography descriptions for Wikipedia tables [Lebret et al., 2016]. The original split of the training / validation / test set is 582,658 / 2,000 / 72,831. We followed the existing works [Chen et al., 2020a; Chen et al., 2020b] to pre-process

this dataset and use 50, 100, 200, and 500 samples from the training dataset as the labeled datasets in the few-shot setting.

We further constructed the unlabeled dataset for each benchmark dataset based on GenWiki [Jin et al., 2020]. This dataset consists of general-domain unpaired structured data and texts sharing the same content distribution. We directly removed the texts in this dataset, and filtered out the structured data that do not have the subjects and objects appearing in the corresponding labeled dataset. Thus, we obtained two unlabeled datasets for WebNLG and WikiBio, respectively, which contain 375,408 / 163,804 samples of structured data without annotated texts[2].

### 4.2 Implementation Detail

In our self-training algorithm, we set the number of curriculum $M_C$ to be 3. For both WebNLG and WikiBio datasets, we split the corresponding unlabeled dataset into 3 subsets which contain the structured data with ($\leq$2) / 3-4 / ($\geq$5) triples. For the hyper-parameters to select pseudo-labeled data, we set $\epsilon_{cov} = 1.0, \epsilon_{gen} = 50\%$. The probabilities of word substitution and triple reordering were set to $p_{word} = p_{triple} = 0.4$.

As for the model structure, we used BART [Lewis et al., 2020] as the text-to-text pre-trained model in our experiments. The base version of BART was adopted because of the limited computational resources. We followed BART to use Byte-Pair Encoding vocabulary with the size of 50,265. The training epoch at each iteration was set to be 20. The learning rate was 0.00003. The batch size was 32 / 24 for WebNLG / WikiBio, respectively. The maximum length of linearized structured data was 256 / 384 for WebNLG / WikiBio, respectively, while the length of text sequences was 128.

### 4.3 Baseline

**Direct Fine-Tuning.** This category of baselines directly fine-tunes the state-of-the-art pre-trained models including KGPT [Chen et al., 2020a], Switch-GPT-2 [Chen et al., 2020b] and BART [Lewis et al., 2020] on the labeled data without the use of unlabeled data. We denoted these baselines as FT-KGPT, FT-Switch-GPT-2 and FT-BART, respectively.

**Task-Adaptive Pre-Training.** This category of baselines designs task-adaptive pre-training methods on unlabeled data before fine-tuning. We chose two representative pre-training tasks: 1) Sequence-level reconstruction (SeqRecon) from BART [Lewis et al., 2020] which decodes complete linearized structured data when encoding corrupted structured data; 2) Graph-level reconstruction (GraphRecon) from JointGT [Ke et al., 2021] which predicts the masked tokens at the output layer of the encoder with the input of corrupted structured data. We denoted them as TAPT-SeqRecon and TAPT-GraphRecon, respectively.

For a fair comparison, we also used BART [Lewis et al., 2020] as the text-to-text pre-trained model for task-adaptive pre-training baselines. We presented the results of our model **CBST** with two ablation models, i.e., **CBST w/o CL** and **CBST w/o CL & Noise**. The former ablation model removes

---

[2]We have conducted triple-level matching on the filtered GenWiki and the test set of WebNLG / WikiBio. The results show that there is no overlap between them.

| % of Training Data | 0.5% | | | 1% | | | 5% | | | 10% | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | B-4 | M | R-L | B-4 | M | R-L | B-4 | M | R-L | B-4 | M | R-L |
| *Direct Fine-Tuning* | | | | | | | | | | | | |
| FT-KGPT | 22.30 | - | - | 25.60 | - | - | 41.20 | - | - | 47.90 | - | - |
| FT-BART | 31.29 | 29.55 | 53.83 | 38.97 | 33.48 | 58.73 | 51.32 | 40.56 | 66.80 | 55.68 | 42.59 | 69.44 |
| *Task-Adaptive Pre-Training* | | | | | | | | | | | | |
| TAPT-SeqRecon | 33.37 | 30.64 | 55.37 | 38.83 | 33.71 | 59.21 | 51.51 | 40.53 | 67.29 | 56.18 | 42.60 | 70.23 |
| TAPT-GraphRecon | 37.47 | 33.31 | 58.84 | 42.29 | 35.93 | 61.22 | 52.25 | 40.78 | 67.96 | 56.57 | 43.02 | 70.44 |
| *Self-Training* | | | | | | | | | | | | |
| CBST (Ours) | 38.74** | 34.24** | 60.28** | 44.40** | 37.37** | 63.62** | 54.98** | 42.18** | 69.44** | 58.78** | 43.94** | 71.30** |
| w/o CL | 37.69 | 34.22 | 59.31 | 43.60 | 37.20 | 62.64 | 53.70 | 41.86 | 68.30 | 57.79 | 43.60 | 70.76 |
| w/o CL & Noise | 37.34 | 34.20 | 58.84 | 43.33 | 37.16 | 62.55 | 52.82 | 41.66 | 67.85 | 56.78 | 43.30 | 70.22 |

Table 2: BLEU-4(B-4), METEOR(M) and ROUGE-L(R-L) in the different settings of WebNLG. The results of FT-KGPT are re-printed from the original paper of KGPT. - means that the results are not reported in the corresponding reference. ** indicates that our model significantly outperforms the best baseline in the corresponding setting (t-test, $p < 0.01$).

| # of Training Data | 50 | 100 | 200 | 500 |
|---|---|---|---|---|
| *Direct Fine-Tuning* | | | | |
| FT-Switch-GPT-2 | 17.20 | 23.80 | 25.40 | 28.60 |
| FT-KGPT | 24.20 | 27.60 | 29.10 | 30.00 |
| FT-BART | 25.83 | 29.38 | 32.53 | 34.80 |
| *Task-Adaptive Pre-Training* | | | | |
| TAPT-SeqRecon | 27.93 | 30.81 | 32.59 | 34.90 |
| TAPT-GraphRecon | 28.86 | 30.40 | 32.49 | 35.24 |
| *Self-Training* | | | | |
| CBST (Ours) | 29.63* | 31.23* | 33.21* | 36.27** |
| w/o CL | 29.05 | 30.70 | 33.13 | 35.03 |
| w/o CL & Noise | 27.96 | 30.54 | 32.65 | 34.87 |

Table 3: BLEU-4 in the different settings of WikiBio. The results of FT-Switch-GPT-2 and FT-KGPT are re-printed from the original paper of KGPT. * indicates that our model significantly outperforms the best baseline in the corresponding setting (t-test, $p < 0.05$), while ** means $p < 0.01$.

| Criterion | Fluency | | | $\kappa$ |
|---|---|---|---|---|
| Model | Win (%) | Lose (%) | Tie (%) | |
| CBST vs. FT-BART | 41.3* | 31.7 | 27.0 | 0.411 |
| CBST vs. TAPT-SeqRecon | 42.0** | 29.3 | 28.7 | 0.410 |
| CBST vs. TAPT-GraphRecon | 41.0* | 32.0 | 27.0 | 0.492 |
| CBST vs. CBST w/o CL | 40.3** | 28.0 | 31.7 | 0.438 |
| CBST vs. CBST w/o CL & Noise | 44.7** | 27.0 | 28.3 | 0.403 |
| Criterion | Adequacy | | | $\kappa$ |
| Model | Win (%) | Lose (%) | Tie (%) | |
| CBST vs. FT-BART | 58.7** | 25.0 | 16.3 | 0.424 |
| CBST vs. TAPT-SeqRecon | 48.0** | 26.3 | 25.7 | 0.427 |
| CBST vs. TAPT-GraphRecon | 44.0* | 32.7 | 23.3 | 0.401 |
| CBST vs. CBST w/o CL | 40.0* | 31.3 | 28.7 | 0.495 |
| CBST vs. CBST w/o CL & Noise | 51.0** | 28.3 | 20.7 | 0.449 |

Table 4: Human evaluation on different methods in the 1% setting of WebNLG. The scores indicate the percentages of win, lose and tie when CBST is compared with other baselines. $\kappa$ is Fleiss' Kappa (all indicate moderate agreement). The scores marked with * mean $p < 0.05$ while ** means $p < 0.01$ in sign test.

curriculum learning from CBST, which is similar to noisy self-training [He *et al.*, 2020]. The latter one simultaneously removes curriculum learning and input noise, which acts as vanilla self-training. All the results were presented with the average values over 3 runs.

### 4.4 Automatic Result

We followed the existing works [Shimorina and Gardent, 2018; Chen *et al.*, 2020a] to adopt BLEU-4 [Papineni *et al.*, 2002], METEOR [Banerjee and Lavie, 2005], and ROUGE-L [Lin, 2004] to evaluate the generated results on WebNLG, and use BLEU-4 as the metric on WikiBio.

The main results on WebNLG and WikiBio are shown in Table 2 and 3. We can observe that CBST significantly outperforms fine-tuning and task-adaptive pre-training methods in all the settings, which shows that our method can explicitly learn the relationship between structured data and texts via self-training and improve the model performance. The comparison between CBST and the ablation models indicates that curriculum learning can alleviate the problem of low-quality pseudo-labeled data and further improve the performance. The noise functions also contribute to the final performance by increasing the smoothness of our model.

### 4.5 Human Evaluation

To further evaluate the quality of generated results, we conducted human evaluation in the 1% setting of WebNLG. We followed the existing works [Ribeiro *et al.*, 2020b] to use two criteria: *fluency* (whether a sentence is grammatically fluent)

and *adequacy* (whether a sentence clearly describes the structured data). We randomly sampled 100 structured data from the test set, and collected the generated results from CBST and other baselines. We adopted pairwise comparison [Ke *et al.*, 2021] between CBST and other baselines. For each pair of generated texts (one from CBST and the other from the corresponding baseline, given the same input structured data), three annotators were hired to determine which text is better (i.e., win, lose or tie) in terms of the above metrics.

Results in Table 4 show that CBST can significantly outperform the baselines based on direct fine-tuning and task-adaptive pre-training in both fluency and adequacy. In addition, the improvement of CBST over two ablation models shows the effectiveness of our curriculum learning module and noise functions to generate fluent texts which describe structured data more clearly. We also calculated Fleiss' Kappa [Fleiss, 1971] for each pairwise comparison to measure the agreement among different annotators, where the results in Table 4 show moderate agreement ($0.4 \le \kappa \le 0.6$).

### 4.6 Ablation Study

We conducted a detailed ablation test to study the effects of different noise functions, selection criteria, and difficulty metrics. We removed each module respectively and presented the results on WebNLG (1%) in Table 5. Note that *w/ DiffLen* denotes that we used the length of linearized structured data as the difficulty metric.

| Model | B-4 | M | R-L |
|---|---|---|---|
| CBST | **44.40** | **37.37** | **63.62** |
| w/o NoiseTriple | 43.87 | 37.24 | 62.92 |
| w/o NoiseWord | 43.53 | 36.93 | 62.94 |
| w/o SelectCov | 43.74 | 37.08 | 63.37 |
| w/o SelectProb | 43.53 | 37.19 | 62.44 |
| w/ DiffLen | 43.93 | 36.75 | 63.27 |

Table 5: Ablation test. NoiseTriple / NoiseWord / SelectCov / SelectProb / DiffLen denotes triple reordering / word substitution / coverage-based selection / probability-based selection / length-based difficulty metric, respectively.

| Model | Hallucination | Missing Fact | Fluency |
|---|---|---|---|
| CBST | **11.3%** | **19.0%** | **4.45** |
| w/o CL | 17.7% | 19.3% | 4.02 |
| w/o CL & Noise | 18.0% | 19.7% | 3.97 |

Table 6: Human evaluation on the pseudo-labeled data generated by different methods at the last iteration.

| $M_C$ | B-4 | M | R-L |
|---|---|---|---|
| 2 | 43.55 | 36.79 | 62.93 |
| 3 | **44.40** | **37.37** | **63.62** |
| 4 | 43.29 | 36.67 | 63.07 |

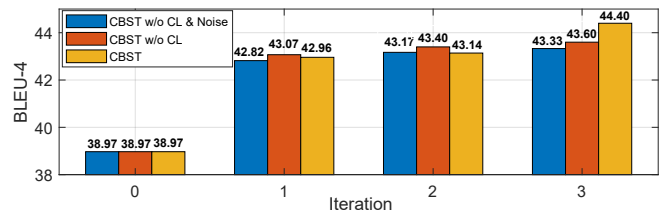Table 7: BLEU-4(B-4), METEOR(M), ROUGE-L(R-L) on WebNLG (1%) with different numbers of curriculum.



Figure 4: BLEU-4 of CBST and two ablation models at different iterations.

Results in Table 5 show that all the modules contribute to the final performance. As for two noise functions, the performance of CBST degrades more in the setting of removing word substitution, which perturbs structured data more flexibly. In terms of the selection criterion, both coverage and generation probability improve the quality of pseudo-labeled data and contribute to the final performance. We can also observe the performance drop when CBST used the length of linearized structured data as the difficulty metric, indicating that the number of input triples is a more proper metric to reflect the difficulty of text generation from structured data.

### 4.7 Analysis on Pseudo-Labeled Data

To study whether our method can improve the quality of pseudo-labeled data via curriculum learning, we evaluated the quality of pseudo-labeled data generated by the teacher model at the last iteration before selection. We resorted to human evaluation since there is no annotated text for unlabeled structured data. We randomly sampled 100 unlabeled structured data and collected the generated results of the teacher models from CBST and the ablation models at the last iteration. Three annotators were hired to judge each generated text from the following fine-grained aspects: 1) *Hallucination*: whether the generated text includes non-existing facts; 2) *Missing fact*: whether the generated text misses input facts; 3) *Fluency*: the fluency score of generated texts (score 1-5 where 5 indicates fully fluent sentences).

We presented the fluency score and the proportions of generated results that belong to hallucination / missing fact in Table 6. Results show that curriculum learning in our method plays an important role in generating fluent and adequate texts to describe unlabeled structured data, resulting in better model performance. The relatively limited improvement on missing facts may be because our base model BART already has the strong ability to generate texts that appear in the input via its pre-training tasks based on text reconstruction.

### 4.8 Analysis on Curriculum Learning

To further analyze how curriculum learning helps self-training in few-shot data-to-text generation, we first demonstrated how the number of curriculum ($M_C$) affects the fi-

nal performance. The results in Table 7 show that the best performance is reached at $M_C = 3$. When $M_C$ is smaller, the teacher model needs to generate texts for unlabeled structured data with multiple triples at early iterations, which may degrade the quality of pseudo-labeled data and the final performance. In contrast, when $M_C$ is larger, the student model is trained on easy unlabeled data for many iterations and cannot utilize the hard cases until the late iterations, which may also affect the model performance.

Furthermore, we set $M_C = 3$ and visualized the performance of the student model at each iteration in Figure 4. Note that the values when the number of iterations equals 0 indicate the performance of direct fine-tuning. At early iterations, the two ablation models perform better because they directly incorporate the whole unlabeled dataset into self-training and acquire a larger pseudo-labeled dataset. However, the improvement of their performance is extremely limited at the second and third iterations since the low-quality texts generated by the teacher model at early iterations may restrict the model performance. For comparison, CBST only utilizes the unlabeled data that satisfy the difficulty metric at each iteration to avoid low-quality pseudo-labeled data. Despite the worse performance at early iterations due to the smaller number of unlabeled data included in the self-training process, CBST still obtains the best performance at the last iteration.

## 5 Conclusion

We introduce self-training to improve text-to-text pre-trained models on few-shot data-to-text generation tasks, which can utilize unlabeled structured data to explicitly model the relationship between structured data and texts. To alleviate the problem of low-quality pseudo-labeled data during self-training, we further propose a novel method called Curriculum-Based Self-Training to rearrange the order of unlabeled data incorporated into self-training based on the difficulty metric. Experimental results show that CBST outperforms fine-tuning and task-adaptive pre-training methods, and achieves state-of-the-art performance in the few-shot setting of WebNLG and WikiBio datasets.

## Acknowledgments

## References

[Banerjee and Lavie, 2005] Satanjeev Banerjee and Alon Lavie. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *Summarization@ACL 2005*, pages 65–72, 2005.

[Bengio *et al.*, 2009] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, volume 382, pages 41–48, 2009.

[Chen *et al.*, 2020a] Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. KGPT: knowledge-grounded pre-training for data-to-text generation. In *EMNLP*, 2020.

[Chen *et al.*, 2020b] Zhiyu Chen, Harini Eavani, Wenhu Chen, Yinyin Liu, and William Yang Wang. Few-shot NLG with pre-trained language model. In *ACL*, 2020.

[Du *et al.*, 2021] Jingfei Du, Edouard Grave, Beliz Gunel, Vishrav Chaudhary, Onur Celebi, Michael Auli, Veselin Stoyanov, and Alexis Conneau. Self-training improves pre-training for natural language understanding. In *NAACL-HLT*, pages 5408–5418, 2021.

[Fleiss, 1971] J. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 1971.

[Gehrmann *et al.*, 2018] Sebastian Gehrmann, Falcon Z. Dai, Henry Elder, and Alexander M. Rush. End-to-end content and plan selection for data-to-text generation. In *INLG*, pages 46–56, 2018.

[Gong *et al.*, 2020] Heng Gong, Yawei Sun, Xiaocheng Feng, Bing Qin, Wei Bi, Xiaojiang Liu, and Ting Liu. Tablegpt: Few-shot table-to-text generation with table structure reconstruction and content matching. In *COLING*, pages 1978–1988, 2020.

[Gururangan *et al.*, 2020] Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *ACL*, pages 8342–8360, 2020.

[He *et al.*, 2020] Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. Revisiting self-training for neural sequence generation. In *ICLR*, 2020.

[Hoyle *et al.*, 2021] Alexander Miserlis Hoyle, Ana Marasovic, and Noah A. Smith. Promoting graph awareness in linearized graph-to-text generation. In *Findings of ACL/IJCNLP*, pages 944–956, 2021.

[Jia *et al.*, 2019] Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. Certified robustness to adversarial word substitutions. In *EMNLP-IJCNLP*, 2019.

[Jin *et al.*, 2020] Zhijing Jin, Qipeng Guo, Xipeng Qiu, and Zheng Zhang. Genwiki: A dataset of 1.3 million content-sharing text and graphs for unsupervised graph-to-text generation. In *COLING*, pages 2398–2409, 2020.

[Kale and Rastogi, 2020] Mihir Kale and Abhinav Rastogi. Text-to-text pre-training for data-to-text tasks. In *INLG*, pages 97–102, 2020.

[Ke *et al.*, 2021] Pei Ke, Haozhe Ji, Yu Ran, Xin Cui, Liwei Wang, Linfeng Song, Xiaoyan Zhu, and Minlie Huang. JointGT: Graph-text joint representation learning for text generation from knowledge graphs. In *Findings of ACL-IJCNLP*, pages 2526–2538, 2021.

[Lebret *et al.*, 2016] Rémi Lebret, David Grangier, and Michael Auli. Neural text generation from structured data with application to the biography domain. In *EMNLP*, pages 1203–1213, 2016.

[Lewis *et al.*, 2020] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, pages 7871–7880, 2020.

[Lin, 2004] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

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

[Raffel *et al.*, 2020] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

[Ribeiro *et al.*, 2020a] Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. Investigating pretrained language models for graph-to-text generation. *CoRR*, abs/2007.08426, 2020.

[Ribeiro *et al.*, 2020b] Leonardo F. R. Ribeiro, Yue Zhang, Claire Gardent, and Iryna Gurevych. Modeling global and local node contexts for text generation from knowledge graphs. *TACL*, 8:589–604, 2020.

[Scudder, 1965] H. J. Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Trans. Inf. Theory*, 11(3):363–371, 1965.

[Shimorina and Gardent, 2018] Anastasia Shimorina and Claire Gardent. Handling rare items in data-to-text generation. In *INLG*, pages 360–370, 2018.

[Trisedya *et al.*, 2018] Bayu Distiawan Trisedya, Jianzhong Qi, Rui Zhang, and Wei Wang. GTR-LSTM: A triple encoder for sentence generation from RDF data. In *ACL*, pages 1627–1637, 2018.

[Xing and Wan, 2021] Xinyu Xing and Xiaojun Wan. Structure-aware pre-training for table-to-text generation. In *Findings of ACL-IJCNLP*, pages 2273–2278, 2021.