

Faithful Low-Resource Data-to-Text Generation through Cycle Training

Zhuoer Wang^{†1} Marcus Collins^{*2} Nikhita Vedula^{*2}
Simone Filice² Shervin Malmasi² Oleg Rokhlenko²

¹Texas A&M University ²Amazon

wang@tamu.edu

{collmr, veduln, filicesf, malmasi, olegro}@amazon.com

Abstract

Methods to generate text from structured data have advanced significantly in recent years, primarily due to fine-tuning of pre-trained language models on large datasets. However, such models can fail to produce output faithful to the input data, particularly on out-of-domain data. Sufficient annotated data is often not available for specific domains, leading us to seek an unsupervised approach to improve the faithfulness of output text. Since the problem is fundamentally one of consistency between the representations of the structured data and text, we evaluate the effectiveness of *cycle training* in this work. Cycle training uses two models which are inverses of each other: one that generates text from structured data, and one which generates the structured data from natural language text. We show that cycle training, when initialized with a small amount of supervised data (100 samples in our case), achieves nearly the same performance as fully supervised approaches for the data-to-text generation task on the WebNLG, E2E, WTQ, and WSQL datasets. We perform extensive empirical analysis with automated evaluation metrics and a newly designed human evaluation schema to reveal different cycle training strategies' effectiveness of reducing various types of generation errors. Our code is publicly available at <https://github.com/EdilLower/CycleNLG>.

1 Introduction

A wealth of information exists in the form of structured knowledge, such as movie information databases or product catalogs, which we may want to verbalize for a variety of purposes, such as comparing two items, or presenting detailed descriptions in a natural language form suitable for conversational assistants. Recent work has tackled this data-to-text generation task using freely available

public datasets, most notably WebNLG (Castro Ferreira et al., 2020) and ToTTo (Parikh et al., 2020). However, there remain two major challenges. First, the volume of training data required for good performance, especially if it is not in a domain represented by one of the existing corpora, is very large. Second, multiple recent papers (Yang et al., 2022; Parikh et al., 2020), *inter alia*, point out that neural natural language generation (NLG) from structured data tends to produce multiple kinds of errors which limit the utility of these models in customer-facing applications. Hallucinations occur when NLG models inject nonsensical words or information not related to the input structured data, into the generated output text. For instance, an NLG model may claim a shirt's color is "three". Simple factual errors occur when an NLG model produces coherent but factually wrong output.

There are two threads of research to consider as we attempt to tackle these problems in the data-to-text setting. The first is designing models that directly produce output more faithful to the input data. The second is designing models to detect and correct factual errors or hallucinations after the output text is generated. In both cases, prior research has generally assumed sufficient pairs of structured data and text as training data to achieve human-level performance on the task. While fact verification models can achieve very high performance, they generally do so when trained on large corpora of 100,000 examples or more. Since performance appears to degrade when evaluated on out-of-domain data (Estes et al., 2022), this presents a significant limitation of fact-verification models. Similarly, corpora like WebNLG contain about 20,000 examples; this is probably too small to achieve human performance even under full supervision (Guo et al., 2020) but is large enough to make it prohibitive to generate domain-specific corpora of the size of WebNLG.

In spite of the above mentioned limitations, very

[†]The research was done during an internship at Amazon.

^{*}These two authors contributed equally to this work.

few of the models developed for data-to-text and table-to-text tasks take advantage of the fact that the task of faithful text generation is fundamentally one of *consistency* between the data and the corresponding text. In fact, despite the WebNLG 2020 challenge being explicitly bi-directional, only three models competing in the challenge leveraged this idea of consistency.

To overcome the aforementioned limitations related to the lack of training data (especially out-of-domain data) and the consistency between structured data and text, we adopt a Cycle Training (Iovine et al., 2022a) approach. We assume *unpaired* data \mathcal{D} , in the form of subject-predicate-object triples, and text \mathcal{T} , which may or may not be from the same domain. We also make use of a small (100 samples) set of paired data and text, $\mathcal{D}_{pr}, \mathcal{T}_{pr}$. Cycle training makes use of two iteratively trained models, a forward model $\mathcal{F} : \mathcal{D} \rightarrow \mathcal{T}$ and a reverse model $\mathcal{R} : \mathcal{T} \rightarrow \mathcal{D}$. Training is unsupervised, namely, we freeze one model and use it to transform one set of inputs, and train the other by using it to predict the original input from the output of the first model. Concretely, in one cycle, we freeze \mathcal{F} , and train \mathcal{R} by reconstructing the input \mathcal{D} as $\mathcal{R}(\mathcal{F}(\mathcal{D}))$. After one training epoch, we reverse the roles of the two models. Remarkably, even though the models are initially quite poor, this can converge to models with near-supervised performance, as we will show. Moreover, we show that this process ensures the *faithfulness* of the output text with respect to the input data, and vice versa, even with very little or no paired data.

We note that a previous data-to-text system, CycleGT, has used cycle training (Guo et al., 2020). We will discuss in detail the differences between CycleGT and our proposed approach in Section 2. Moreover, we examine in detail the conditions under which cycle training works well, with an emphasis on domains and the nature of the training text and structured data. We find that unsupervised cycle training outperforms low-resource fine-tuned models and can achieve near fully-supervised performance when initialized and post-tuned with a small amount of annotated data. We detail the results and findings in Section 5. Thus, to build on past research in self-consistent data-to-text generation, we make these novel contributions:

(i) We successfully apply cycle training to both the data-to-text and text-to-data models using only a pre-trained language model, T5, without recourse

to graph methods or other auxiliary models.

(ii) We show that cycle training achieves nearly the same performance as supervised models for some domains.

(iii) We present an extensive empirical analysis on the conditions under which cycle training works well, and on the data-to-text faithfulness with respect to different types of generation errors.

(iv) We design a novel counting and ranking based annotation schema to more comprehensively evaluate the faithfulness of the generated text from the standpoints of correctness, faithfulness, data coverage, and fluency. Our schema improves upon the rating-based schema used for the WebNLG 2020 Challenge, in terms of objectiveness, consistency, precision and ease of evaluation.

2 Related Work

Multiple data-to-text and table-to-text tasks have been presented in the literature, such as WebNLG (Gardent et al., 2017a; Colin et al., 2016; Gardent et al., 2017b), DART (Nan et al., 2020), ToTTo (Parikh et al., 2020), and WikiTableT (Chen et al., 2021), which primarily consist of data from general-purpose sources like Wikipedia. Several large language models (Herzig et al., 2020; Liu et al., 2021; Yang et al., 2022) have been trained on large scale table-to-text corpora (Chen et al., 2019) to perform fact verification. However, these models may not perform well on specific domains they have not been trained on, such as e-commerce (Estes et al., 2022; Vedula et al., 2022). Therefore, we must either find a way to easily generate new data to train large data-to-text models, or use unsupervised methods. Recently, Xiang et al. (2022) attempted to augment training data using GPT-3 (Brown et al., 2020), and Su et al. (2021) employed an information retrieval system to build prototypes for the generation. Our work makes orthogonal contributions to these studies, as we directly utilize the underlying unpaired data and text of a target corpus without recourse to any additional information retrieval or generation systems. Further, the above-mentioned data-to-text tasks have been evaluated primarily on automatic word- or n-gram-level metrics such as BLEU (Papineni et al., 2002) or METEOR (Banerjee and Lavie, 2005), with minimal (and mostly subjective) evaluation of faithfulness. In this work, we design a novel annotation schema to perform a more comprehensive evaluation of the faithfulness of the generated text

to the input data.

Cycle training (Zhu et al., 2017; Zhou et al., 2016) relies on two models which are essentially inverse transforms of each other that are used to create “cycles”, which should return identical output to the input given. There are two distinct forms of cycle training. The first form (Zhou et al., 2016) aims to learn to transform from one input form to another, e.g., to learn rotations of a car in one image to another. The second is the use of a “cycle consistency loss” as an auxiliary loss to some other task, e.g., in generative adversarial networks performing style transfer on images (Zhu et al., 2017). NLG typically relies on models which are auto-regressive and non-differentiable. This precludes the direct use of cycle consistency losses (Guo et al., 2020; Pang and Gimpel, 2019; Iovine et al., 2022a). Nonetheless, we can still use cycle training via an alternating training strategy where we freeze one model and train the other, and vice versa (Lample et al., 2017; Pang and Gimpel, 2019). In this work, we train solely using cycle consistency. Cycle training has been recently applied to language processing tasks. In one text-to-text application, Iovine et al. (2022b) use a similar unsupervised methodology to perform bidirectional text transformations for converting keyword search queries to natural language questions, and *vice versa*. It has also been used for Named Entity Recognition in the absence of large annotated text (Iovine et al., 2022a). In this case, one model extracts entities, and the inverse model creates text from those entities. The approach is limited by the fact that there are many ways to realize sentences with the same entities. Put differently, there is no strong requirement of cycle consistency, and this will become even more apparent as we analyze the conditions under which cycle training works well in data-to-text tasks.

To the best of our knowledge, the only work to explicitly call out the self-consistency requirement of data-to-text generation tasks is the CycleGT model (Guo et al., 2020) developed for data-to-text generation on the WebNLG dataset. One key advantage of cycle training is that it need not rely on any supervision, and instead relies primarily or solely on the self-consistency of inputs and outputs. However, CycleGT relies on a pre-existing NER model to extract entities from the output text. The authors then train an inverse model to predict the links between entities and predicates. Should the entities not be recognized by their NER system, the

model will fail overall; this is not an uncommon situation in applications such as online shopping (Estes et al., 2022; Vedula et al., 2023), where entities are complex or change frequently (Malmasi et al., 2022). In principle, a separate NER model could be built using cycle training, as in CycleNER (Iovine et al., 2022a), but the CycleGT authors did not do so. In this work, we design a simple approach using pre-trained language generation models, fine-tuned for both data-to-text and text-to-data generation cycles.

3 Methodology

3.1 Backbone Models

The pre-requisite of cycle training is having two mutually inverse models. We adopt T5, an evidently strong-performing model according to the WebNLG 2020 challenge (Castro Ferreira et al., 2020; Agarwal et al., 2020; Guo et al., 2020), as our backbone model for both forward generation, ($\mathcal{F} : \mathcal{D} \rightarrow \mathcal{T}$ that performs RDF-to-text generation) and reverse generation, ($\mathcal{R} : \mathcal{T} \rightarrow \mathcal{D}$ that performs text-to-RDF generation). T5 is a large sequence-to-sequence model pre-trained with the unsupervised span-mask denoising objective and several supervised text generation tasks like summarization and translation (Raffel et al., 2020). We linearize the RDF triples of each sample into a sequence d that denotes the subject, predicate, and object of each triple by the [S], [P], and [O] tags respectively. Therefore, both RDF-to-text and text-to-RDF can be treated and trained as sequence-to-sequence generation tasks. We further train or optionally fine-tune the T5 backbone models, as detailed in Section 4, with the teacher forcing (Williams and Zipser, 1989; Lamb et al., 2016) learning objective for task-specific generation. This means that for the training of the auto-regressive decoder, we do not propagate the model decoded next token but force each input to be the correct gold token for training.

3.2 Cycle Training of the Backbone Models

Iterative Back-Translation (IBT) (Hoang et al., 2018) has been reported as an effective training schema that enforces cycle consistency for various NLP tasks (Guo et al., 2020; Iovine et al., 2022a). We apply this idea to iteratively cycle train our models. This consists of the Data-Text-Data (DTD) cycle that enforces the self-consistency of data, and the Text-Data-Text (TDT) cycle that similarly en-

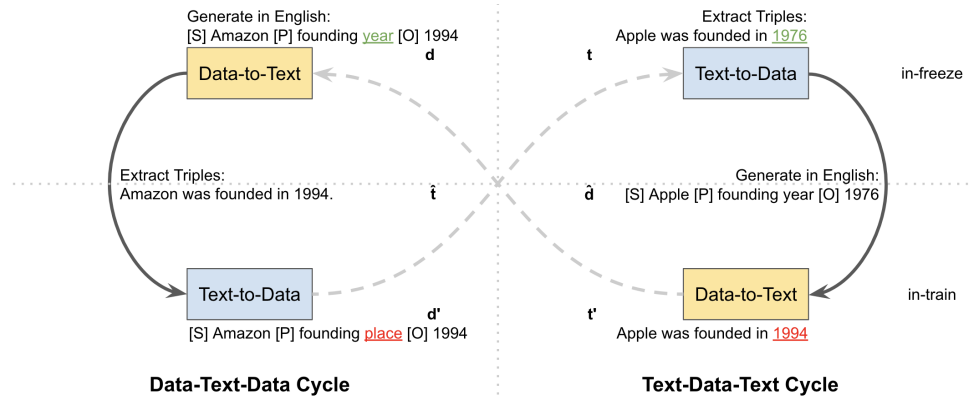


Figure 1: Cycle Training of the Data-to-Text model and Text-to-Data model. For each cycle, the upper-level models are frozen to generate the intermediate text for the training of the lower-level models, that attempt to reconstruct the initial inputs (d, t denote initial inputs of the upper-level models; \hat{t}, \hat{d} denote the upper-level models’ generations that serve as inputs to the lower-level models; d', t' denote the generations of the lower-level models).

forces the self-consistency of text. As shown in Figure 1, for the DTD cycle, the Data-to-Text model takes the linearized triples d as input and generates the associated intermediate text \hat{t} . Sequentially, the Text-to-Data model is trained with the objective of reconstructing d with the supplied \hat{t} . The reconstruction loss $\mathcal{L}_{d'}$ is the averaged negative log likelihood shown below where d_i denotes the i -th token of sequence t and $|d|$ is the sequence length:

$$\mathcal{L}_{d'} = -\frac{1}{|d|} \sum_{i=0}^{|d|} \log p(d_i | d_0, \dots, d_{i-1}, \hat{t})$$

In a reverse manner, for the TDT cycle, the Text-to-Data model first takes text t as input and generates the associated linearized triples \hat{d} . Sequentially, the Text-to-Data model is trained with the objective of reconstructing t with the supplied \hat{d} . The reconstruction loss $\mathcal{L}_{t'}$ is the averaged negative log likelihood shown below where t_i denotes the i -th token of sequence t and $|t|$ is the sequence length:

$$\mathcal{L}_{t'} = -\frac{1}{|t|} \sum_{i=0}^{|t|} \log p(t_i | t_0, \dots, t_{i-1}, \hat{d})$$

Due to the non-differentiable procedure of generating discrete intermediate outputs of tokens, the reconstruction loss can only propagate through the second model of each cycle, namely the Text-to-Data model of the DTD cycle and the Data-to-Text model of the TDT cycle. Therefore, the training of the two models can only proceed with the alternation of the TDT cycle and the DTD cycle so that both models’ performance may gradually improve.

4 Experimental Setup

4.1 Data and Baselines

We experiment on existing data sources that have annotated pairs of data triples and reference texts.

WebNLG (Colin et al., 2016; Gardent et al., 2017b; Castro Ferreira et al., 2020) is a well-established dataset that has supported multiple challenges on four tasks: RDF-to-English (Text), RDF-to-Russian (Text), English (Text)-to-RDF, and Russian (Text)-to-RDF. Each WebNLG sample consists of a set of subject-predicate-object triples and up to three associated human-written reference texts that faithfully express and verbalize the information contained in the triple set. We use the English data from the most recent 3.0 version of the WebNLG corpus, from the WebNLG+ 2020 challenge.

DART (Nan et al., 2020) is a large-scale data-to-text dataset that unifies and builds upon multiple data resources including E2E (Novikova et al., 2017), WikiSQL (WSQL) (Zhong et al., 2017), WikiTableQuestions (WTQ) (Pasupat and Liang, 2015), and WebNLG (Gardent et al., 2017a). To better facilitate our experiments and evaluations on different domains, we separately utilize the human-annotated portion of E2E, WTQ, and WSQL from DART. To align the data formats in accordance with WebNLG, we also drop some WSQL and WTQ samples that contain non-conventional structural tags. The DART dataset hereafter refers to the cleaned, WebNLG-excluded, and human-annotated portion of E2E, WTQ, and WSQL.

Table 1 shows detailed dataset statistics. When the data is used for cycle training, we follow previous work and split all the paired samples into one separate corpus of shuffled text, and another separate corpus of shuffled triple sets. For the linearized sequences, as shown in Figure 1, we: (1) prefix the string “Generate in English:” to the input sequence of the RDF-to-text model and pre-

Dataset	Domain	Split Size (Train/Dev/Test)	Unique Predicates	Triples/Sample (median/max)	Vocab Size	Tokens/Sample (median/max)
WebNLG	DBpedia (16 categories)	35,426/4,464/7,305	1,236	3 / 7	20,126	21 / 80
E2E	Restaurants	33,482/1,475/1,475	41	4 / 7	6,158	22 / 73
WTQ	Wikipedia (open-domain)	3,253/361/155	5,013	2 / 10	11,490	13 / 107
WSQL	Wikipedia (open-domain)	526/59/38	946	2 / 6	2,353	12 / 34

Table 1: Datasets statistics and comparison.

fix the string “Extract Triples:” to the input of the text-to-RDF model; (2) convert camel-cased or snake-cased subjects, predicates and objects to regular strings; and (3) normalize accented characters.

Fine-tuning large pre-trained language models, such as BERT (Devlin et al., 2019), BART (Lewis et al., 2020), and T5 (Raffel et al., 2020) has been proven to be effective in achieving new state-of-the-art performance on numerous tasks. Fine-tuning refers to the supplemental training of a pre-trained model on a dataset of the target task and domain. We detail and perform the following three baseline fine-tuning strategies in this work:

Fully supervised fine-tuning: We fine-tune T5 with the entire in-domain (with respect to the test set) data as the supervised baseline.

Low-resource fine-tuning: We fine-tune the T5-base model with 100 randomly selected sets of triples and their associated reference texts to formalize a low-resource supervised baseline. We deem 100 annotated samples to be a small enough amount, that is easily achievable with a relatively low human annotation effort.

Low-resource fine-tuning with additional pre-training: When using text from the target domain for cycle training, the teacher forcing algorithm naturally raises the probability of generating the target domain tokens, which may result in performance gains in token matching metrics (Section 5.1). To study the influence of using in-domain text, we further pre-train the T5 model with in-domain text and an unsupervised span-mask denoising objective prior to the low-resource fine-tuning process. As our main objective is to probe a training strategy orthogonal to the model structure, we only include the above three baselines to control the model structure, data pre-requisites, and parameter sizes.

4.2 Comparing Cycle Training Strategies and Pre-requisites

We explore two different training strategies evaluating the effectiveness and generalizability of cycle training under different data constraints.

Unsupervised cycle training: As the most constrained low-resource scenario, in unsupervised cycle training we directly employ the IBT schema to cycle-train the forward model and reverse model with unpaired text and triple sets in turns.

Low-resource cycle training: In this setting, a small amount of paired text and triple sets are accessible. For fair comparison and consistency, we utilize the same subset of data as the low-resource fine-tuning baseline described in Section 4.1. The low-resource paired data is leveraged through *pre-cycle fine-tuning*, which first trains the forward and reverse model with the paired data before employing the IBT schema to cycle-train the two models.

Guo et al. (2020) and Iovine et al. (2022a) vaguely state that the latent content or entity distribution of the text corpus and the data corpus must have some uncertain degree of overlap to make the cycle training approach work. To empirically assess this pre-requisite condition, we apply unsupervised cycle training with the same size of text and data corpus at different matching levels, as a rough approximation of overlap of the latent content or entity distribution. Specifically, we randomly select half of the WebNLG triplets as the data corpus. We purposefully select five equal-sized text corpora that contain 0%, 25%, 50%, 75%, and 100% of the originally related reference text; and complementarily include 100%, 75%, 50%, 25%, 0% of unrelated reference text respectively.

4.3 Training Parameters

We use the T5-base model which has 12 layers, a hidden size of 768, 12 self-attention heads, and 220M parameters. We use the AdamW optimizer with linear weight decay, a max input length of 256, a learning rate of $3e-4$, and an effective batch size of 256. At inference time, we decode with the beam search algorithm using 4 beams and a generation length varying between 3 tokens and 256 tokens. We train each model up to 50 epochs with a delta of 0.05 basis points and a patience of 5 epochs as the early stopping criteria. We select the best model by

Method	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	BLEU	BertScore	PARENT
Tested on WebNLG							
Fully-supervised fine-tuning	59.99 _(0.10)	40.93 _(0.18)	49.32 _(0.15)	39.76 _(0.04)	42.83 _(0.21)	95.41 _(0.02)	45.67 _(0.30)
Low-resource fine-tuning	55.55 _(0.67)	36.63 _(0.37)	46.21 _(0.35)	35.22 _(0.70)	33.63 _(0.87)	94.60 _(0.08)	41.37 _(0.54)
+ additional pre-training	55.28 _(0.43)	35.71 _(0.32)	45.41 _(0.24)	35.26 _(0.46)	33.44 _(0.59)	94.33 _(0.06)	39.47 _(0.52)
Unsupervised cycle training	58.65 _(0.53)	37.70 _(1.02)	46.18 _(0.59)	37.98 _(0.33)	36.36 _(2.35)	94.42 _(0.26)	43.24 _(1.10)
Low-resource cycle training	60.21 _(0.21)	40.56 _(0.42)	48.71 _(0.17)	39.74 _(0.32)	41.77 _(0.70)	95.18 _(0.04)	46.14 _(0.36)
Tested on E2E							
Fully-supervised fine-tuning	69.77 _(0.10)	42.87 _(0.17)	50.93 _(0.18)	52.90 _(0.43)	29.35 _(0.47)	94.76 _(0.02)	41.91 _(0.61)
Low-resource fine-tuning	66.62 _(0.15)	39.68 _(0.25)	48.59 _(0.18)	48.80 _(0.39)	25.31 _(0.31)	94.35 _(0.02)	39.56 _(1.21)
+ additional pre-training	66.88 _(0.40)	39.45 _(0.33)	48.65 _(0.36)	50.11 _(0.65)	26.29 _(0.55)	94.35 _(0.04)	39.65 _(0.53)
Unsupervised cycle training	63.43 _(0.81)	37.73 _(0.32)	45.96 _(0.61)	50.49 _(0.78)	27.92 _(0.37)	93.71 _(0.09)	37.97 _(0.30)
Low-resource cycle training	69.53 _(0.25)	42.48 _(0.20)	50.51 _(0.28)	53.02 _(0.24)	29.22 _(0.12)	94.74 _(0.02)	41.39 _(0.70)
Tested on WTQ							
Fully-supervised fine-tuning	62.25 _(0.66)	34.59 _(0.61)	49.41 _(0.57)	39.17 _(0.86)	21.18 _(0.53)	92.88 _(0.05)	24.18 _(0.74)
Low-resource fine-tuning	55.89 _(0.88)	31.60 _(0.81)	46.73 _(0.64)	31.98 _(0.57)	15.34 _(0.72)	91.91 _(0.14)	23.36 _(1.05)
+ additional pre-training	55.57 _(0.68)	30.48 _(0.80)	44.47 _(0.74)	33.73 _(0.74)	15.89 _(0.39)	91.53 _(0.17)	22.88 _(0.43)
Unsupervised cycle training	61.27 _(0.50)	33.45 _(0.52)	48.22 _(0.44)	39.06 _(0.22)	20.46 _(0.69)	92.67 _(0.04)	23.05 _(0.35)
Low-resource cycle training	61.54 _(0.29)	34.25 _(0.78)	49.07 _(0.45)	39.09 _(0.60)	20.93 _(0.98)	92.66 _(0.10)	24.39 _(0.84)
Tested on WSQL							
Fully-supervised fine-tuning	58.27 _(1.79)	32.77 _(1.15)	48.40 _(2.44)	37.95 _(0.99)	22.97 _(1.38)	93.18 _(0.19)	24.00 _(2.07)
Low-resource fine-tuning	56.37 _(1.15)	31.60 _(0.59)	49.42 _(0.77)	33.57 _(0.24)	23.34 _(1.03)	92.57 _(0.18)	23.68 _(1.11)
+ additional pre-training	56.01 _(0.66)	30.92 _(0.92)	47.00 _(1.18)	35.34 _(0.86)	21.18 _(0.65)	92.24 _(0.33)	22.66 _(0.56)
Unsupervised cycle training	42.24 _(0.23)	15.17 _(0.13)	33.52 _(0.23)	29.45 _(0.29)	4.03 _(0.15)	85.37 _(0.14)	14.63 _(0.17)
Low-resource cycle training	58.71 _(1.43)	33.13 _(1.90)	51.01 _(1.43)	37.43 _(1.04)	25.60 _(1.58)	93.03 _(0.18)	25.84 _(1.42)

Table 2: Evaluation of data-to-text generation (**bold**: best of all; underlined: best of low-resource settings). We report the average and standard deviation (in parenthesized subscripts) of each metric for 5 repeated runs.

the validation set’s METEOR score - the ranking metric of the WebNLG 2020 challenge, and we report the aforementioned model’s performance on the test set. We repeat each experiment 5 times with different random seeds and report the average and standard deviation of each metric.

5 Results and Discussion

5.1 Automatic Evaluation

We assess each system/strategy with five widely-used automatic metrics that measure the generation quality from three different aspects: token-matching, semantic similarity, and faithfulness.

ROUGE (Lin, 2004) is a recall-oriented metric that calculates the overlapping n-grams (ROUGE-N for N-grams) and word sequences (ROUGE-L) between the reference text and generated text.

BLEU (Papineni et al., 2002) is a precision-oriented metric calculating overlapping n-grams between the reference text and generated text.

METEOR (Banerjee and Lavie, 2005) computes the unigram match between the reference text and generated text based on the tokens’ surface form, stemming, synonyms, and paraphrase similarities.

BertScore (Zhang et al., 2020) measures the semantic similarity of the reference text and generated text via the utilization of the contextual embeddings from BERT for the calculation of the cosine similarity of best-matching token pairs.

PARENT (Dhingra et al., 2019) is an entailment-based token-matching metric that calculates the F1 score based on entailed precision (an n-gram is correct if it occurs in the reference text or entailed by the input data) and entailed recall (recall against the reference text input data, adjusted by a weight parameter). It measures the faithfulness of the generated text with respect to the input data.

Table 2 displays the performance of multiple data-to-text generation approaches under various settings. We observe that unsupervised cycle training generally falls short of the fully-supervised fine-tuning method’s performance. When compared with the low-resource fine-tuning method, it scored higher on WebNLG and WTQ but performed worse on E2E and WSQL, where the performance gap on WSQL is larger. We attribute such divergence to the difference in the number of unique predicates and vocabulary. Cycle training should be able to improve the model’s generalizability and robustness through exposure to larger amounts of diverse text and structured data, and through its capability of gradually learning different data-to-text associations. For datasets like E2E and WSQL, their smaller vocabulary size and number of unique predicates imply that a small amount of annotated samples might cover a great deal of the datasets’ underlying variation. This leads to a strong low-resource fine-tuning performance that has smaller

Overlapping Level	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	BLEU	BertScore	PARENT
0%	52.50 _(0.43)	31.16 _(0.40)	40.14 _(0.46)	35.99 _(0.46)	26.69 _(1.03)	92.59 _(0.12)	34.33 _(0.58)
25%	56.23 _(0.67)	34.59 _(0.82)	43.46 _(0.63)	37.23 _(0.17)	32.21 _(1.74)	93.63 _(0.22)	39.28 _(0.96)
50%	58.64 _(0.34)	37.40 _(0.60)	46.05 _(0.41)	38.07 _(0.28)	35.83 _(1.07)	94.41 _(0.17)	43.09 _(0.68)
75%	58.64 _(0.32)	37.66 _(0.26)	46.36 _(0.23)	37.78 _(0.18)	36.91 _(0.37)	94.46 _(0.09)	43.47 _(0.37)
100%	58.75 _(0.28)	38.04 _(0.44)	46.44 _(0.19)	37.86 _(0.25)	37.39 _(0.79)	94.57 _(0.12)	43.76 _(0.32)

Table 3: Cycle training with the same amount of data at different overlapping levels. We report the average and standard deviation (in parenthesized subscripts) of each metric for 5 repeated runs.

performance gaps with the fully-supervised counterparts, and overshadows the unsupervised cycle training method.

However, when a small amount of annotated data is made available for initializing the cycle training, the low-resource cycle training strategy significantly improves the generation performance over the low-resource fine-tuning method, and achieves competitive performance with respect to the fully-supervised method. Such an improvement is consistent across all four datasets and five types of evaluation metrics. Notably, when applied to multi-domain and open-domain datasets (WebNLG, WTQ, and WSQL), low-resource cycle training generated texts that have better faithfulness to the input data, evident from the PARENT score, compared to the fully-supervised fine-tuning approach. Compared with the setting that applies additional pre-training, it is evident that cycle training works beyond simply raising the probability of generating target domain tokens.

As for the experiments on cycle training with unpaired datasets at different overlapping levels, the results in Table 3 show that performance sharply increases at the beginning with the increase of overlapping levels and then turns to flatten at around the 50% overlapping level. This suggests that when the size is the same, the unpaired data corpus and text corpus used for cycle training need to have at least 50% entities (or say, latent information) overlap to achieve performance at an ideal level. We deem 50% as a reasonable level since many related but unpaired texts and structured data (e.g., content and infoboxes from Wikipedia, product specification tables and descriptions from online shopping platforms, etc.) may have higher information overlap. Hence, based on our experimental results, we believe that low-resource cycle training is a universally applicable approach that can effectively learn from vast unpaired structured data and texts with minimal human effort.

5.2 Human Evaluation

To quantitatively compare generated text with respect to correctness, faithfulness, data coverage, and fluency, we develop a new counting and ranking-based annotation schema, and use it to conduct human evaluation. Our schema features better objectiveness, consistency, and precision compared to the 0-100 rating-based schema used for the WebNLG 2020 Challenge. We define the following measures (full annotation guidelines, including disambiguation examples, and screenshots of the annotation interface available in Appendix A):

Count of Factual Errors (FE) measures the factual correctness of the generated text with respect to the entities (subject and object) and predicates of the input triplets. Factual errors are information in the generations that contradict the information in the input subject-predicate-object context. For each attempted predicate given in the input triplets, the annotator is asked to increase the factual error count if the subject and/or object of the predicate’s associated expression doesn’t match facts from the input.

Count of Hallucination Errors (HE) measures the relevance of the generated text with respect to the input triplets. Hallucination errors occur when words or phrases in the generation cannot be inferred from the input subject-predicate-object triplets, for instance, because the value does not make logical sense, or because the predicate of the expression is not present in any triple. Unlike FEs, HEs add information not present in the triplets or reference, but do not directly contradict the triplets. The annotator is asked to increase the HE count if a piece of information contained in the generated text is not presented in, or cannot be *reasonably inferred* by the input triplets. For better consistency and less ambiguity, a *reasonable inference* is defined as a piece of information contained in the generated text that isn’t present in the input triplets but is present in the reference text.

Count of Information Misses (IM) measures the information coverage of the generated text with

Method	FE	HE	IM	FP
Combined				
Low-resource fine-tuning	8.05	14.84	21.39	2.00
Low-resource cycle-training	0.49	2.57	3.36	1.80
Fully-supervised fine-tuning	2.08	11.48	8.46	1.73
WebNLG				
Low-resource fine-tuning	6.72	7.21	15.90	1.91
Low-resource cycle-training	0.00	1.47	1.82	1.89
Fully-supervised fine-tuning	0.00	6.72	10.29	1.73
E2E				
Low-resource fine-tuning	0.00	1.18	6.43	1.99
Low-resource cycle-training	0.00	0.00	0.84	1.86
Fully-supervised fine-tuning	0.00	0.00	0.00	1.64
WTQ				
Low-resource fine-tuning	14.71	15.69	33.82	2.16
Low-resource cycle-training	0.00	0.00	1.96	1.75
Fully-supervised fine-tuning	8.33	24.51	8.82	1.85
WSQL				
Low-resource fine-tuning	10.78	35.29	29.41	1.93
Low-resource cycle-training	1.96	8.82	8.82	1.72
Fully-supervised fine-tuning	0.00	14.71	14.71	1.76

Table 4: Normalized Human Evaluation Results (lower is better; **bold**: best of all.)

respect to the predicates given in the input triplets. For each predicate given in the input triplets, the annotator is asked to increase the IM count if the generated text does not attempt to express the predicate.

Fluency Preference (FP) measures the quality of the generated text in terms of the grammar, structure, and coherence of the text. The annotator is asked to compare the fluency of pairs of generated texts within a batch, to compile the final ranking that reflects the annotator’s subjective preference. The fluency comparison and ranking only considers the grammar, structure, and coherence of the text independent of IM, FE, and HE.

In terms of the training time required to perform

the task accurately, we collected the error annotations (FE, HE, IM) from two domain experts and the fluency annotations from crowd-sourced workers respectively via an annotation tool built on the Appen¹ platform. To enforce the annotation quality and foster future research on explainable automatic error analysis, we ask the domain experts to mark the token(s) that constitute an FE or HE, and to select the triple(s) that constitute the IM before counting the respective errors. The domain experts independently annotate the same set of 204 randomly sampled generations with a resulting agreement (Cohen’s kappa score (Artstein and Poesio, 2008)) of 0.74 for FE, 0.69 for HE, and 0.85 for IM, which is very satisfactory given the complexity of the task. For the relatively more subjective fluency ranking task, we use the average of three crowd-sourced native English speakers’ judgments for each generation. As generating longer text for larger triple sets is more difficult than generating for smaller triplets, we normalize the counts of FE, HE, and IM by the number of their input triples. Therefore, the FE, HE, and IM we report in Table 4 can be interpreted as the probability of making such errors per input data triple. We show an example of our error analysis in Table 5, and provide additional examples in Appendix B.

Our human evaluation suggests that low-resource cycle training consistently reduces factual errors, hallucination errors and information misses. From Section 5.1, cycle training presents a larger performance gain when applied to datasets that have more variations in terms of underlying relations and surface realizations. When looking together with Table 2, the human evaluation of errors and information coverage correlates better with the PARENT score, which confirms PARENT’s capability of measuring faithfulness. It is also evident from the annotation results that all three evaluated data-to-text generation models are more likely to make hallucination errors over factual errors, which calls for more future effort to alleviate hallucinations. In terms of the generated texts’ fluency, low-resource cycle training is able to improve over the low-resource fine-tuning method but still cannot consistently beat the fully-supervised approach.

6 Conclusions

In this work, we demonstrated the application of cycle training for data-to-text generation. We sys-

¹<https://appen.com/>

Input Triple Set	1. [S] Liselotte Grschebina [P] birthplace [O] Karlsruhe 2. [S] Liselotte Grschebina [P] nationality [O] Israel 3. [S] Liselotte Grschebina [P] training [O] School of Applied Arts in Stuttgart 4. [S] Karlsruhe [P] country [O] Germany 5. [S] Israel [P] language [O] Modern Hebrew	
Reference Text	- Liselotte Grschebina was an Israeli who attended the school of Applied Arts in Stuttgart. She was born in Karlsruhe, Germany. Israel's national language is Modern Hebrew. - Liselotte Grschebina is an Israeli national who did her studies at the School of Applied Arts in Stuttgart and was born in Karlsruhe, which is in Germany unlike Israel whose language is Modern Hebrew. - Liselotte Grschebina was born in Karlsruhe which is located in Germany, has Israeli nationality (the language in Israel is Modern Hebrew), and studied at the School of Applied Arts in Stuttgart.	
Method	Generation	Remark
Fully-supervised fine-tuning	Liselotte Grschebina was born in Karlsruhe, Germany. She studied at the School of Applied Arts in Stuttgart. Modern Hebrew is spoken in Israel.	1 IM
Low-resource fine-tuning	Liselotte Grschebina was born in Karlsruhe, Israel and trained at the School of Applied Arts in Stuttgart.	3 IMs, 1 FE
Low-resource cycle-training	Liselotte Grschebina was born in Karlsruhe, Germany and was a student at the School of Applied Arts in Stuttgart. Modern Hebrew is spoken in Israel.	1 IM

Table 5: Error analysis example.

tematically investigated the effectiveness of cycle training across different domains, and the application of pre-cycle fine-tuning in low-resource settings. We showed that our approach substantially improved data-to-text generation performance in low-resource settings, achieved competitive performance compared to fully-supervised models, and also improved the faithfulness of the generated text through a reduction in factual errors, hallucinations and information misses, even when compared to fully supervised approaches. We also designed a schema for effective human evaluation of data-to-text generation, that improves upon prior work and encourages more objective and consistent reviews of faithfulness.

Limitations

We recognize that our annotation and analysis methods can require considerable human labor, that can limit the amount of annotated data we can collect. Also, despite cycle training being generally accepted as a model-agnostic approach, we were not able to test a wide variety of backbone models due to resource constraints. In addition, though we relaxed the entity constraints and made cycle training for data-to-text generation end-to-end, the non-differentiability problem remains unsolved. The intermediate outputs generated by the first model of each cycle are assumed to be correct. This is a weak assumption that may propagate misleading training signals to the second model of each cycle, particularly in the early stage of the training.

To address these limitations, future work may focus on the following directions: 1) building differentiable cycle training models; 2) exploring au-

tomated error detection methods and building models that may utilize such signals; and 3) assessing different backbone models, including large language models like GPT-X, with the cycle training approach.

Acknowledgements

First and foremost, we extend our appreciation to Prof. James Caverlee for his unwavering support that was vital for the completion of this work. We gratefully acknowledge the contributions of the following individuals for their expert advice as well as their participation in our preliminary human annotation study, which helped us a lot in refining our experiments, annotation guidelines and annotation interface: Dr. Giuseppe Castellucci, Dr. Besnik Fetahu, Prof. Eugene Agichtein, Dr. Saar Kuzi, Jason Ingyu Choi, Dr. Zhiyu Chen, Dr. Tuan M. Lai, Lingbo Mo, and Yicheng Wang. We also would like to express our gratitude to the three reviewers and the meta reviewer for their constructive suggestions.

References

- Oshin Agarwal, Mihir Kale, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2020. [Machine translation aided bilingual data-to-text generation and semantic parsing](#). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 125–130, Dublin, Ireland (Virtual). Association for Computational Linguistics.
- Ron Artstein and Massimo Poesio. 2008. [Survey article: Inter-coder agreement for computational linguistics](#). *Computational Linguistics*, 34(4):555–596.

- Satanjeev Banerjee and Alon Lavie. 2005. [METEOR: An automatic metric for MT evaluation with improved correlation with human judgments](#). In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Thiago Castro Ferreira, Claire Gardent, Nikolai Ilinykh, Chris van der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina. 2020. [The 2020 bilingual, bi-directional WebNLG+ shared task: Overview and evaluation results \(WebNLG+ 2020\)](#). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 55–76, Dublin, Ireland (Virtual). Association for Computational Linguistics.
- Mingda Chen, Sam Wiseman, and Kevin Gimpel. 2021. [WikiTableT: A Large-Scale Data-to-Text Dataset for Generating Wikipedia Article Sections](#). *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 193–209.
- Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyong Zhou, and William Yang Wang. 2019. [TabFact: A Large-scale Dataset for Table-based Fact Verification](#). In *International Conference on Learning Representations (ICLR)*, arXiv, Addis Ababa, Ethiopia.
- Emilie Colin, Claire Gardent, Yassine Mrabet, Shashi Narayan, and Laura Perez-Beltrachini. 2016. [The WebNLG Challenge: Generating Text from DBpedia Data](#). *Proceedings of the 9th International Natural Language Generation conference*, pages 163–167.
- 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, Minneapolis, Minnesota. Association for Computational Linguistics.
- Bhuvan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William W Cohen. 2019. [Handling Divergent Reference Texts when Evaluating Table-to-Text Generation](#). *arXiv*. This is the PARENT evaluation metric paper.
- Alex Estes, Nikhita Vedula, Marcus Collins, Matthew Cecil, and Oleg Rokhlenko. 2022. [Fact Checking Machine Generated Text with Dependency Trees](#). *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017a. [Creating Training Corpora for NLG Micro-Planners](#). *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188.
- Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017b. [The WebNLG Challenge: Generating Text from RDF Data](#). *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133.
- Qipeng Guo, Zhijing Jin, Xipeng Qiu, Weinan Zhang, David Wipf, and Zheng Zhang. 2020. [CycleGT: Unsupervised graph-to-text and text-to-graph generation via cycle training](#). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 77–88, Dublin, Ireland (Virtual). Association for Computational Linguistics.
- Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. [TaPas: Weakly Supervised Table Parsing via Pre-training](#). *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333.
- Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. [Iterative back-translation for neural machine translation](#). In *Proceedings of the 2nd workshop on neural machine translation and generation*, pages 18–24.
- Andrea Iovine, Anjie Fang, Besnik Fetahu, Oleg Rokhlenko, and Shervin Malmasi. 2022a. [CycleNER: An Unsupervised Training Approach for Named Entity Recognition](#). *Proceedings of the ACM Web Conference 2022*, pages 2916–2924.
- Andrea Iovine, Anjie Fang, Besnik Fetahu, Jie Zhao, Oleg Rokhlenko, and Shervin Malmasi. 2022b. [CycleKQR: Unsupervised bidirectional keyword-question rewriting](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11875–11886, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. 2016. [Professor forcing: A new algorithm for training recurrent networks](#). *Advances in neural information processing systems*, 29.

- Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc’Aurelio Ranzato. 2017. Unsupervised Machine Translation Using Monolingual Corpora Only. *arXiv*.
- 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.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2021. TAPEX: Table Pre-training via Learning a Neural SQL Executor. *arXiv*.
- Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022. SemEval-2022 task 11: Multilingual complex named entity recognition (MultiCoNER). In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1412–1437, Seattle, United States. Association for Computational Linguistics.
- Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangu Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Mutethia Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chern Tan, Xi Victoria Lin, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. 2020. DART: Open-Domain Structured Data Record to Text Generation. *arXiv*.
- Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206, Saarbrücken, Germany. Association for Computational Linguistics.
- Richard Yuanzhe Pang and Kevin Gimpel. 2019. Unsupervised Evaluation Metrics and Learning Criteria for Non-Parallel Textual Transfer. *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 138–147.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. ToTTo: A Controlled Table-To-Text Generation Dataset. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186.
- Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, Beijing, China. Association for Computational Linguistics.
- 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. *J. Mach. Learn. Res.*, 21(1).
- Yixuan Su, Zaiqiao Meng, Simon Baker, and Nigel Collier. 2021. Few-shot table-to-text generation with prototype memory. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 910–917, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Nikhita Vedula, Marcus Collins, Eugene Agichtein, and Oleg Rokhlenko. 2022. What matters for shoppers: Investigating key attributes for online product comparison. In *European Conference on Information Retrieval*, pages 231–239. Springer.
- Nikhita Vedula, Marcus Collins, Eugene Agichtein, and Oleg Rokhlenko. 2023. Generating explainable product comparisons for online shopping. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 949–957.
- Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.
- Jiannan Xiang, Zhengzhong Liu, Yucheng Zhou, Eric Xing, and Zhiting Hu. 2022. ASDOT: Any-shot data-to-text generation with pretrained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1886–1899, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Jingfeng Yang, Aditya Gupta, Shyam Upadhyay, Luheng He, Rahul Goel, and Shachi Paul. 2022. TableFormer: Robust Transformer Modeling for Table-Text Encoding. *arXiv*. Very interesting approach to use scalar attention biases between different types of content, e.g. table columns and the input query.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

- Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*.
- Tinghui Zhou, Philipp Krähenbühl, Mathieu Aubry, Qixing Huang, and Alexei A. Efros. 2016. [Learning Dense Correspondence via 3D-Guided Cycle Consistency](#). *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 117–126.
- Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. [Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks](#). *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251.

Appendix

A Annotation Guidelines

In this section, we include descriptions of the human annotation task performed in this work.

For this annotation task, the annotators will be provided a set of input triplets in the subject-predicate-object structure, and the annotators will be asked to provide their judgement of four model-generated text snippets associated with the input triplets. Our target is to annotate the 1) Count of Factual Errors, 2) Count of Hallucination Errors, 3) Count of Information Misses, and 4) Fluency Preference for the generations. We use two different Appen interface-pages: one for the annotation of the three types of error counts, and one for the annotation of Fluency Preference.

A.1 Annotation of Error Counts

A.1.1 Count of Factual Errors (FE)

Count of Factual Errors (FE) measures the factual correctness of the generated text with respect to the entities (subject and object) and predicates of the input triplets.

Annotation Instruction: Factual errors are information in the generations which contradict the information in the subject-predicate-object context. For each attempted predicate given in the input triplets, the annotator is supposed to increase the count if [the subject and/or object of the predicate’s associated expression *does not match the facts* suggested by the input triplets].

Examples: (See Table 6)

A.1.2 Count of Hallucination Errors (HE)

Count of Hallucination Errors (HE) measures the relevance of the generated text with respect to the input triplets.

Annotation Instruction: Hallucination errors occur when words or phrases in the generation cannot be inferred from the subject-predicate-object triplets, for instance because the value doesn’t make logical sense, or because the predicate of the expression isn’t present in any triple. Distinguished from FEs, HEs invent information not in the triplets or reference, but do not directly contradict the triplets. The annotator is supposed to increase the count if [a piece of information contained in the generated text *is not presented in or can not be reasonably inferred by* the input triplets]. For better consistency and less ambiguity, reasonable inference is defined as a piece of information

contained in the generated text isn’t presented in the input triplets but is presented in the reference text.

Examples: (See Table 7)

A.1.3 Count of Information Misses (IM)

Count of Information Misses (IM) measures the information coverage of the generated text with respect to the predicates given in the input triplets.

Annotation Instruction: For each predicate given in the input triplets, the annotator is supposed to increase the count by 1 if [the generated text *did not attempt* to express the predicate].

Examples: (See Table 8)

A.1.4 Annotation Interface for Errors

The annotation task is presented batch-by-batch. Each batch contains one shared input triplet and three model-generated text snippets (in random order) with respect to the input triplets. The annotators will see the input triplets data and the reference ground-truth data at first. Please keep in mind that the ground-truth data is just a reference for the convenience of better understanding the input triplets and the boundary of "reasonable inference" and they may not be perfect. To begin with, we ask the annotators to provide token level annotations of FE and HE. The “Context” is the input triplets shown before. The annotators can click the [grey-rounded i] button at the upper-right corner to see information regarding the use of the annotation tool. The annotators can also click the [grey-rounded i] button next to the tag to see a recap of its definition. Annotations of overlapped tokens are permitted. After finishing up the token-level FE and HE annotation, please provide the count of FE and the count of HE respectively. Next, the annotators need to identify if there’s any missed information in the generation. If "Yes", the annotators will be asked to check the IMs. See Figure 2 and Figure 3 for screenshots of the annotation interface for FE, HE, and IM.

A.1.5 Fluency Preference (FP)

Fluency Preference (FP) measures the quality of the generated text in terms of the grammar, structure, and the coherence of the text.

Annotation Instruction: The annotator is supposed to perform pairwise fluency comparison of the generated texts within a batch to compile the final ranking that reflects the annotator’s subjective preference. The fluency comparison and ranking

Input Triple Set 1	<ol style="list-style-type: none"> 1. [S] Mexico [P] currency [O] Mexican peso 2. [S] Mexico [P] demonym [O] Mexicans 3. [S] Bionico [P] course [O] Dessert 4. [S] Bionico [P] ingredient [O] Raisin 5. [S] Bionico [P] country [O] Mexico
Generations and Reasonings	<ul style="list-style-type: none"> • 1 FE: Bionico is a dessert made with Raisin and Mexican peso. It is a dish from Mexico. <ul style="list-style-type: none"> – According to the input data, Mexican peso is the currency of Mexico not the ingredient of Bionico, so it is a FE. • 2 FEs: In Mexico, the currency is the Mexican peso. It is a dessert with a Raisin ingredient. <ul style="list-style-type: none"> – "It" is a pronoun that grammatically refers to Mexican peso, so the subjects of attempted expressions for triplet 3 and 4 are wrong, which results in two FEs. • 1 FE: Bionico is the demonym of Raisin <ul style="list-style-type: none"> – This is considered as an attempt to express triplet 2 but is factually incorrect.
Input Triple Set 2	<ol style="list-style-type: none"> 1. [S] Alan B. Miller Hall [P] address [O] 101 Ukrop Way 2. [S] Alan B. Miller Hall [P] height [O] 36.5 meters
Generations and Reasonings	<ul style="list-style-type: none"> • 2 FEs: Alan B. Miller Hall located at 440 Terry Avenue has a height of 365 meters. <ul style="list-style-type: none"> – Although 440 Terry Avenue and 365 may seem like hallucinations, they counter the fact that the address of Alan B. Miller Hall is 101 Ukrop Way and the fact that the Hall's height is 36.5 meters. We consider them as FEs instead of HEs because the input data explicitly contradicts these generated strings (which is how FEs are defined).

Table 6: Disambiguation examples of Factual Errors (FE).

Count of Hallucination Errors (HE)	
Input Triple Set 1	<ol style="list-style-type: none"> 1. [S] ALCO RS-3 [P] build date [O] May 1950 - August 1956 2. [S] ALCO RS-3 [P] power type [O] Diesel-electric transmission 3. [S] ALCO RS-3 [P] builder [O] Montreal Locomotive Works 4. [S] ALCO RS-3 [P] length [O] 17068.8
Reference Text	<ul style="list-style-type: none"> - The ALCO RS-3 was produced between May 1950 and August 1956 and was built by Montreal Locomotive Works. This locomotive has a diesel-electric transmission and is 17068.8 millimetres in length. - The ALCO RS-3 was produced between May 1950 and August 1956 and was built by Montreal Locomotive Works. It has a diesel-electric transmission and is 17068.8 millimetres long. - The ALCO RS-3, built by the Montreal Locomotive Works between May 1950 and August 1956, has a diesel-electric transmission and measures 17068.8 millimetres in length.
Generations and Reasonings	<ul style="list-style-type: none"> • 1 HE: The Montreal Locomotive Works built the ALCO RS-3 from May 1950 - August 1956. It has a diesel-electric transmission and a length of 17068.8 meters. <ul style="list-style-type: none"> – The unit expression of <i>meters</i> is considered as a HE since such information doesn't appear in the input data or the reference text (hence not considered as a reasonable inference). • 0 HE: The ALCO RS-3 was built by the Montreal Locomotive Works between May 1950 and August 1956. It has a diesel-electric transmission and is 17068.8 millimetres long. <ul style="list-style-type: none"> – The unit expression of <i>millimeters</i> doesn't appear in the input data but appears in the reference text (hence it is considered as a reasonable inference), so it is not a HE.
Input Triple Set 2	<ol style="list-style-type: none"> 1. [S] Liselotte Grschebina [P] death place [O] Israel 2. [S] Liselotte Grschebina [P] death place [O] Petah Tikva 3. [S] Israel [P] population density [O] 387.63 4. [S] Israel [P] long name [O] State of Israel 5. [S] Liselotte Grschebina [P] nationality [O] Israel
Reference Text	<ul style="list-style-type: none"> - Liselotte Grschebina is an Israeli national who died in Petah Tikva, Israel which is formally known as the State of Israel and has a population density of 387.63 people per square kilometre of land area. - Liselotte Grschebina was an Israeli who died in Petah Tikva, Israel which has a population density of 387.63 people per square kilometre of land area and is named "State of Israel." - Liselotte Grschebina has Israeli nationality and died in Petah Tikva, Israel. Israel has the population density of 387.63 and its full name is the State of Israel.
Generations and Reasonings	<ul style="list-style-type: none"> • 1 HE: Liselotte Grschebina was born in Israel and died in Petah Tikva. Israel has a population density of 387.63 people. <ul style="list-style-type: none"> – The birth place information doesn't appear in the input data and cannot be reasonably inferred either, so it is considered as a HE.

Table 7: Disambiguation examples of Hallucination Errors (HE).

Input Triple Set 1	<ol style="list-style-type: none"> 1. [S] Liselotte Grschebina [P] birth place [O] Karlsruhe 2. [S] Liselotte Grschebina [P] nationality [O] Israel 3. [S] Liselotte Grschebina [P] training [O] School of Applied Arts in Stuttgart 4. [S] Karlsruhe [P] country [O] Germany 5. [S] Israel [P] language [O] Modern Hebrew
Generations and Reasonings	<ul style="list-style-type: none"> • 1 IM: Liselotte Grschebina was born in Karlsruhe, Germany. She studied at the School of Applied Arts in Stuttgart. Modern Hebrew is spoken in Israel. <ul style="list-style-type: none"> – Triplet 2 hasn't been expressed. – The expression of a predicate can be implicit. For instance, <i>Karlsruhe, Germany</i> is an implicit expression with respect to <i>triplet 4</i>. • 2 IMs: Liselotte Grschebina was born in Karlsruhe, Israel and trained at the School of Applied Arts in Stuttgart. <ul style="list-style-type: none"> – Triple 2 and 5 haven't been expressed. – <i>Karlsruhe, Israel</i> can be considered as an expression attempt of triplet 4 although it contains factual errors. IM only counts information coverage with respect to the predicates and neglects entities (subject/object). • 0 IM: Liselotte Grschebina was born in Karlsruhe, Germany and studied at the School of Applied Arts in Stuttgart. She is Israeli and speaks Modern Hebrew. <ul style="list-style-type: none"> – <i>(She/Liselotte) speaks Modern Hebrew</i> can be considered as an expression attempt of triplet 5. <i>Somebody (Israeli) speaks Modern Hebrew</i> is a reasonable alternative expression attempt of <i>the language in Israel is Modern Hebrew</i>.
Input Triple Set 2	<ol style="list-style-type: none"> 1. [S] Liselotte Grschebina [P] death place [O] Israel 2. [S] Liselotte Grschebina [P] death place [O] Petah Tikva
Generations and Reasonings	<ul style="list-style-type: none"> • 1 IM: Liselotte Grschebina died in Petah Tikva. <ul style="list-style-type: none"> – This is a special case which we count as having a IM. In rare cases, the predicates in the input data may look the same due to omissions. Here, the predicate of triplet 1 is actually <i>death place (country)</i> and of triplet 2 is actually <i>death place (city)</i>. Hence, this generation only expresses one triplet's predicate.

Table 8: Disambiguation examples of Information Misses (IM).

shall only consider the *grammar*, *structure*, and the *coherence* of the text **without** the consideration of IM, FE, and HE.

Examples: Since FP is a relatively more subjective measure that asks for overall preference, we only provide some contrasting examples for the three aspects of fluency.

- **Grammar:** Generation A is better than B because B is grammatically incorrect/influent.
 - Generation A: 108, written by karen maser, has 2.12 million U.S. viewers.
 - Generation B: 108 U.S. viewers million is 2.12, written by karen maser.
- **Structure:** Generation A is better than B because the pieces of information in A are more naturally connected and expressed.
 - Generation A: Andrew Rayel is a member of the Bobina band that plays trance music.
 - Generation B: Andrew Rayel is an associated band/associated musical artist with Bobina. His genre is Trance music.
- **Coherence:** Generation A is better than B because *She speaks modern Hebrew* is more logically and consistently connected with the pre-

vious sentences compared to *Modern Hebrew is spoken in Israel*.

- Generation A: Liselotte Grschebina was born in Karlsruhe, Germany and trained in the School of Applied Arts in Stuttgart. She speaks modern Hebrew.
- Generation B: Liselotte Grschebina was born in Karlsruhe, Germany. She studied at the School of Applied Arts in Stuttgart. Modern Hebrew is spoken in Israel.

A.1.6 Annotation Interface for FP

The annotators may see two to three generations, and the annotators are asked to perform pairwise comparison and rank the generations by their grammar, structure, and coherence without considering information coverage and factual errors. The annotators should start with 1 for the highest-ranked/most-fluent text of the generations within the batch. Ranking tie is permitted, but note this is a ranking task, so the annotators will need to check the numbers in a normal ranking manner. If the annotators see two generations [A, B], and A is better than B, then the annotators should select 1 for A and 2 for B instead of 3 for B. If the annotators see three generations [A, B, C], and A is identical to B, B is better than C, then the annotators should select 1 for A, 1 for B, 3 for C instead of 2 for C. See Fi-

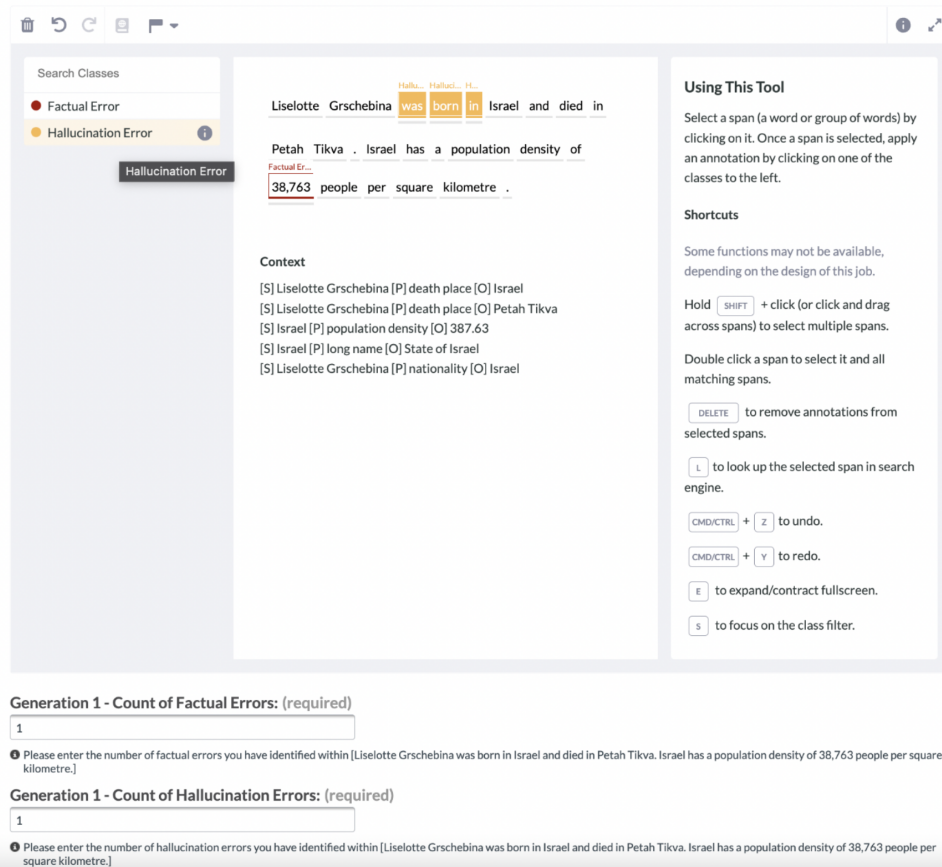


Figure 2: Annotation Interface for FE and HE.

Is there any missed information in Generation 1? (required)

Yes

No

Generation 1 - Information Misses: (required)

[S] Liselotte Grschebina [P] death place [O] Israel

[S] Liselotte Grschebina [P] death place [O] Petah Tikva

[S] Israel [P] population density [O] 387.63

[S] Israel [P] long name [O] State of Israel

[S] Liselotte Grschebina [P] nationality [O] Israel

Please check any unattempted predicates within [Liselotte Grschebina was born in Israel and died in Petah Tikva. Israel has a population density of 38,763 people per square kilometre.]

Figure 3: Annotation Interface for IM.

Please rank the following 2 pieces of text with respect to their grammar, sentence structure, and the coherence of the text. Assign a score of 1 to the most fluent text, and 3 to the least fluent text.

The Waterman is an Italian pub in the city centre near Raja Indian Cuisine. It is not family-friendly.

Text 1 Ranking (required)

1	2	3
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

The Waterman is an Italian pub located in the city centre near Raja Indian Cuisine. It is not family-friendly.

Text 2 Ranking (required)

1	2	3
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 4: Annotation Interface for FP.

Figure 4 for a screenshot of the annotation interface for FP.

B Additional Result Analysis for WebNLG and DART

In this section, we show some additional samples of text generated by multiple models trained under different cycle settings, and the errors made by these models for the WebNLG dataset in Table 9 and DART dataset in Table 10.

WebNLG - Sample 1		
Input Triple Set	1. [S] Liselotte Grschebina [P] death place [O] Israel 2. [S] Liselotte Grschebina [P] death place [O] Petah Tikva 3. [S] Israel [P] population density [O] 387.63 4. [S] Israel [P] long name [O] State of Israel 5. [S] Liselotte Grschebina [P] nationality [O] Israel	
Reference Text	- Liselotte Grschebina is an Israeli national who died in Petah Tikva, Israel which is formally known as the State of Israel and has a population density of 387.63 people per square kilometre of land area. - Liselotte Grschebina was an Israeli who died in Petah Tikva, Israel which has a population density of 387.63 people per square kilometre of land area and is named "State of Israel". - Liselotte Grschebina has Israeli nationality and died in Petah Tikva, Israel. Israel has the population density of 387.63 and its full name is the State of Israel.	
Method	Generation	Remark
Fully-supervised fine-tuning	Liselotte Grschebina was an Israeli national who died in Petah Tikva, Israel. The long name of Israel is the State of Israel and the population density is 387.63 inhabitants per square kilometre.	
Low-resource fine-tuning	Liselotte Grschebina was born in Israel and died in Petah Tikva. Israel has a population density of 387,63 people.	3 IMs, 1 FE, 1HE
Low-resource cycle-training	Liselotte Grschebina died in Petah Tikva, Israel. The population density of Israel is 387.63 inhabitants per square kilometre and the long name is the State of Israel.	1 IM
WebNLG - Sample 2		
Input Triple Set	1. [S] Alan B. Miller Hall [P] architect [O] Robert A. M. Stern 2. [S] Alan B. Miller Hall [P] address [O] 101 Ukrop Way 3. [S] Alan B. Miller Hall [P] current tenants [O] Mason School of Business 4. [S] Alan B. Miller Hall [P] completion date [O] 2009-06-01 5. [S] Alan B. Miller Hall [P] location [O] Virginia	
Reference Text	- The Mason School of Business is located at the Alan B. Miller Hall in Virginia at 101 Ukrop Way. The architect of the building was Robert A M Stern and the building completion date was 01/06/2007. - The address of Alan B. Miller Hall is 101 Ukrop Way, Virginia and the hall is currently tenanted by The Mason School of Business. The hall was designed by Robert A.M. Stern and was completed on June 1st, 2009. - Alan B. Miller Hall, which was designed by Robert A.M. Stern is in the State of Virginia, at 101 Ukrop Way. The building was finished on 01/06/2009 and is currently tenanted by The Mason School of Business.	
Method	Generation	Remark
Fully-supervised fine-tuning	The Mason School of Business are the current tenants of Alan B Miller Hall which was designed by Robert A M Stern and completed on 1 June 2009 at 101 Ukrop Way, Virginia.	
Low-resource fine-tuning	Alan B. Miller Hall was completed on June 1, 2009 and is located in Virginia. The architect is Robert A. M. Stern.	2 IMs
Low-resource cycle-training	Alan B Miller Hall in Virginia was designed by Robert A M Stern and is located at 101 Ukrop Way. The current tenants are the Mason School of Business. It was completed on 1 June 2009.	
WebNLG - Sample 3		
Input Triple Set	1. [S] Bootleg Series Volume 1: The Quine Tapes [P] producer [O] The Velvet Underground 2. [S] Bootleg Series Volume 1: The Quine Tapes [P] genre [O] Rock music 3. [S] The Velvet Underground [P] genre [O] Proto-punk	
Reference Text	- The rock album "Bootleg Series Volume 1: The Quine Tapes" was produced by proto-punk band The Velvet Underground. - The Bootleg Series Volume I: The Quine tapes is a rock album, by the proto-punk band, The Velvet Underground. - Bootleg Series Volume 1: The Quine Tapes is a rock album that also fits into the proto-punk genre. It was produced by the Velvet Underground.	
Method	Generation	Remark
Fully-supervised fine-tuning	The Velvet Underground, whose genre is proto-punk, produce the bootleg series volume 1: The Quine Tapes. The genre of the bootleg series is rock music.	
Low-resource fine-tuning	The Velvet Underground is the producer of Bootleg Series Volume 1: The Quine Tapes. The genre of the bootleg series is rock music.	1 IM
Low-resource cycle-training	The Velvet Underground is the producer of Bootleg Series Volume 1: The Quine Tapes. The band performs rock music and their genre is Proto-punk.	

Table 9: Error analysis of additional data samples generated by various cycle training settings, for the WebNLG dataset.

DART - Sample 1		
Input Triple Set	1. [S] The Vaults [P] eat type [O] restaurant 2. [S] The Vaults [P] food [O] French 3. [S] The Vaults [P] pricerange [O] moderate 4. [S] The Vaults [P] area [O] riverside 5. [S] The Vaults [P] family friendly [O] yes 6. [S] The Vaults [P] near [O] Raja Indian Cuisine	
Reference Text	- The French restaurant, The Vaults, is a moderately priced, kid friendly restaurant in riverside near Raja Indian Cuisine.	
Method	Generation	Remark
Fully-supervised fine-tuning	The Vaults is a moderately priced French restaurant in the riverside area near Raja Indian Cuisine. It is kid friendly.	
Low-resource fine-tuning	The Vaults is a moderately priced French restaurant located near Raja Indian Cuisine.	2 IMs
Low-resource cycle-training	The Vaults is a French restaurant near Raja Indian Cuisine in the riverside area. It is kid friendly and moderately priced.	
DART - Sample 2		
Input Triple Set	1. [S] K-2 1000 m [P] silver [O] Bulgaria Berenike Faldum Daniela Nedeva 2. [S] K-2 1000 m [P] gold [O] Germany Anne Knorr Debora Niche 3. [S] K-2 1000 m [P] bronze [O] Hungary Aliz Sarudi Erika Medveczky	
Reference Text	- Anne Knorr won gold, Berenike Faldum won silver, and Aliz Satudi won bronze.	
Method	Generation	Remark
Fully-supervised fine-tuning	Bulgaria, Berenike Faldum, Daniela Nedeva won silver, Germany Anne Knorr, Debora Niche won gold, Hungary Aliz Sarudi, Erika Medveczky won bronze.	
Low-resource fine-tuning	Bulgaria Berenike Faldum Daniela Nedeva won gold in the K-2 1000 m.	2 IMs 1 FE
Low-resource cycle-training	Bulgaria Berenike Faldum Daniela Nedeva won silver, Germany Anne Knorr Debora Niche won gold, Hungary Aliz Sarudi and Erika Medveczky won bronze in K-2 1000 m.	
DART - Sample 3		
Input Triple Set	1. [S] Illinois 2 [P] result [O] Lost re-election Republican gain 2. [S] Barratt O'Hara [P] first elected [O] 1948 3. [S] Illinois 2 [P] candidates [O] Richard B. Vail R 53.6% Barratt O'Hara D 46.4% 4. [S] Illinois 2 [P] incumbent [O] Barratt O'Hara 5. [S] Barratt O'Hara [P] party [O] Democratic	
Reference Text	- Incumbent barratt o'hara was first elected in 1948.0.	
Method	Generation	Remark
Fully-supervised fine-tuning	Barratt O'Hara was first elected in 1948.0.	4 IMs
Low-resource fine-tuning	Barratt O'Hara was first elected in 1948.	4 IMs
Low-resource cycle-training	In Illinois 2, Barratt O'Hara lost re-election Republican gain and was first elected in 1948.	3 IMs
DART - Sample 4		
Input Triple Set	1. [S] Clowns [P] eat type [O] pub 2. [S] Clowns [P] price range [O] more than £30 3. [S] Clowns [P] customer rating [O] 3 out of 5 4. [S] Clowns [P] near [O] All Bar One	
Reference Text	- For a 3 out of 5 star customer rated pub near All Bar One, Clowns is relatively expensive.	
Method	Generation	Remark
Fully-supervised fine-tuning	Clowns is a pub near All Bar One. It has a customer rating of 3 out of 5 and a price range of more than £30.	
Low-resource fine-tuning	Clowns is a pub with a customer rating of 3 out of 5	2 IMs
Low-resource cycle-training	Clowns is a pub near All Bar One. It has a customer rating of 3 out of 5 and a price range of more than £30.	

Table 10: Error analysis of additional data samples generated by various cycle training settings, for the DART dataset.

ACL 2023 Responsible NLP Checklist

A For every submission:

- A1. Did you describe the limitations of your work?
Section 7
- A2. Did you discuss any potential risks of your work?
Not applicable. Our work is based on publicly available datasets and the method won't introduce additional risks.
- A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1
- A4. Have you used AI writing assistants when working on this paper?
Left blank.

B Did you use or create scientific artifacts?

Section 4 and Section 5

- B1. Did you cite the creators of artifacts you used?
Section 4 and Section 5
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
The artifacts we used are publicly available and accessible. The license can be found on their respective GitHub or project page.
- B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 4 and Section 5
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. We use existing datasets that have no personal or offensive information.
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 4
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 4

C Did you run computational experiments?

Section 4 and Section 5

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

- C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4
- C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4 and Section 5
- C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
We used the default settings that are consistent with previous work
- D** **Did you use human annotators (e.g., crowdworkers) or research with human participants?**
Section 5
- D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Section 5
- D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Section 5
- D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
We didn't discuss it in the paper but the use of data was made clear to the annotators
- D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Ethics review is not required due to the nature of our data
- D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
The demographic and geographic characteristics are irrelevant to our work, and such information was not collected from the annotators.