R2D2: Robust Data-to-Text with Replacement Detection

Linyong Nan **Lorenzo Jaime Yu Flores** **Yilun Zhao** **Yixin Liu**
Luke Benson **Weijin Zou** **Dragomir Radev**
Yale University
{linyong.nan, lj.flores, yilun.zhao}@yale.edu

Abstract

Unfaithful text generation is a common problem for text generation systems. In the case of Data-to-Text (D2T) systems, the factuality of the generated text is particularly crucial for any real-world applications. We introduce R2D2, a training framework that addresses unfaithful Data-to-Text generation by training a system both as a generator and a faithfulness discriminator with additional replacement detection and unlikelihood learning tasks. To facilitate such training, we propose two methods for sampling unfaithful sentences. We argue that the poor entity retrieval capability of D2T systems is one of the primary sources of unfaithfulness, so in addition to the existing metrics, we further propose named entity based metrics to evaluate the fidelity of D2T generations. Our experimental results show that R2D2 systems could effectively mitigate the unfaithful text generation, and they achieve new state-of-the-art results on FeTaQA, LogicNLG, and ToTTo, all with significant improvements.

1 Introduction

Data-to-Text generation is the task of generating a text sequence that describes some salient information of a knowledge source. Unlike Text-to-Text generation whose input source is a text sequence containing knowledge that is not extracted and represented in the canonical structured format, we assume that the input of a Data-to-Text system is represented in some structured format, e.g., RDF (Gardent et al., 2017), relational or entity tables (Lebret et al., 2016; Wiseman et al., 2017). The Data-to-Text task can be divided into two distinct components as in many other text generation tasks (Reiter and Dale, 2000; Gatt and Krahmer, 2018). The first component involves selecting salient information from the structured knowledge either based on natural language query or other indication of saliency, and the second component comprises organizing and planning of the previous selections to allow realization of the surface text. Although this task has been studied comprehensively in many works, from task design, modeling techniques, to application in different domains (Gardent et al., 2017; Lebret et al., 2016; Wiseman et al., 2017; Novikova et al., 2017; Parikh et al., 2020; Nan et al., 2022), existing Data-to-Text (D2T) systems exhibit a shortcoming that cannot be neglected: they fail to reliably generate sentences that are faithful given the salient content of the input table (Chen et al., 2020a,b, 2021a; Uehara et al., 2020; Ji et al., 2022). This limitation prevents the application of D2T systems in real world scenarios. We therefore need to investigate possible remedies.

We introduce a framework that prevents unfaithful Data-to-Text generation by training a Data-to-Text system both as a generator as well as a faithfulness discriminator. For faithfulness discrimination, we adopt the *replaced token detection* objective, which was first proposed in ELECTRA (Clark et al., 2020). It was applied to the pre-training stage of the large-scale language models for more sample-efficient training of contextualized representations of sentences. ELECTRA is tasked to discriminate between original natural sentences and token-replaced sentences by locating the positions of replacement. The replaced tokens are sampled from a proposal distribution using a generator such as a Masked Language Model to fill some masked tokens.

In our work, we perturbed the entailed reference sentences with two different methods, a knowledge-based one and a model-based one, to obtain unfaithful sentences whose surface forms are close to those of original sentences (therefore having similar sequence likelihoods), but contradict to the input table. Then we investigated ways of incorporating the discrimination task into the existing maximum likelihood learning. Specifically, we explored the settings of learning the sentence-level detection and generation in tandem, and the token-level

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 6903–6917
December 7-11, 2022 ©2022 Association for Computational Linguistics

detection and generation in tandem. In addition, we also experiment with incorporating the unlikelihood training objective (Welleck et al., 2019) on these unfaithful sentences to test its utility.

We conduct experiments on three Data-to-Text datasets to test the general applicability of our approach: FeTaQA (Nan et al., 2022), LogicNLG (Chen et al., 2020a), and ToTTo (Parikh et al., 2020). Each dataset presents distinct challenges while faithful generation is a common problem. We find that adding the faithfulness discrimination task mitigates the unfaithful Data-to-Text generation, supported by our results on multiple datasets, on all of which we are able to achieve new state-of-the-art results with evident improvements. We compare and analyze the performance of our system and existing state-of-the-art systems. To ensure the validity of the comparison, we also evaluate various metrics for their aptness of faithfulness evaluation. We released our model and code at `https://github.com/Yale-LILY/r2d2`.

2 Method

2.1 Preliminaries

The de facto Data-to-Text machine learning task requires conditional language modeling of the sequence pair $X = (x_1, \ldots, x_M), Y = (y_1, \ldots, y_N)$ using a neural model parameterized with θ: $p_\theta(y_1, \ldots, y_N | x_1, \ldots, x_M)$, where $(y_1, \ldots, y_N)$ is a natural language sentence that faithfully describes the salient part of the input data which is linearized, along with other contexts such as query or metadata, into the sequence $X = (x_1, \ldots, x_M)$.

We want to sample unfaithful sentences that are in the vicinity of surface forms of reference sentences, therefore also likely to be generated when only learning with maximum likelihood loss. We aim to examine the effectiveness of our proposed discrimination objectives in guiding the D2T model to attain **separable representations** for these superficially similar but factually critical sentences, and more importantly, we investigate how generation can benefit from these additional objectives for robustness. We call our method **R**obust **D**ata-to-Text with **R**eplacement **D**etection (`R2D2`), because we assign the Data-to-Text model both a generation task and a discrimination task (replacement detection). This process is illustrated in Figure 1.

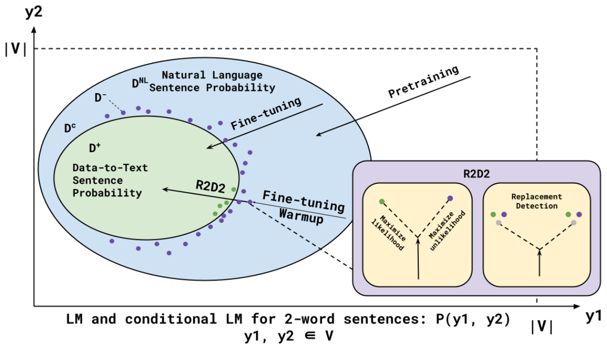


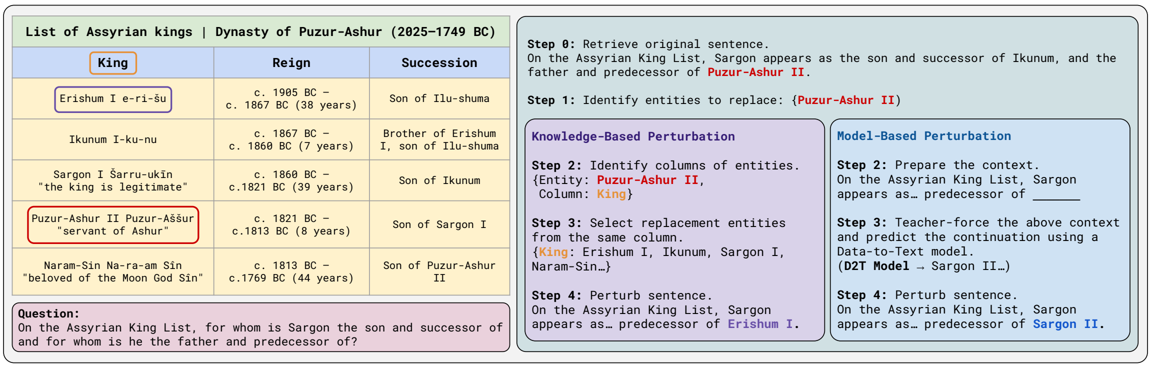
Figure 1: Illustration of how R2D2 fine-tuning fits the probability distribution of faithful sentences shown in a Data-to-Text task, assume the length of sentence is 2 for the visualization purpose.

Many existing works that study the unfaithful text generation problem in summarization and translation have investigated the source of inconsistencies between the system output and the input (Cao et al., 2017; Maynez et al., 2020; Goyal and Durrett, 2020, 2021; Chen et al., 2021a). The main source of unfaithfulness is that the outputs contain facts that cannot be entailed (necessary consequence) from any explicitly stated facts or inferences derived from the input table. As we represent facts with subject-predicate-object triples, these contradictions originate from wrong predictions of entities (subject or object), predicates, or wrong arrangements. This motivates our proposal of different methods of obtaining unfaithful sentences in Section 2.2. Then we describe two learning objectives that we proposed to add to the Data-to-Text modeling: 1) replacement detection objective in Section 2.3; and 2) unlikelihood objective in Section 2.4. In Section 2.5, we formulate our `R2D2` fine-tuning that leverages these two objectives in addition to the standard negative log likelihood loss for robust training of a Data-to-Text model.

2.2 Faithfulness-based Replacement

We aim to obtain sentences that are not entailed from the input table by replacing entities or predicates in the original sentences. It is challenging to reliably extract the predicates in canonical form that can be compared and replaced with one another, but it is feasible to extract and compare entities from the system generations and the input table. Therefore we replace a span of tokens that constitute an entity in the original sentence with another candidate entity comprising one or more tokens. We adopted a RoBERTa-large-based named entity recognizer[1] to extract all the entities in a sentence.

[1] `https://huggingface.co/Jean-Baptiste/roberta-large-ner-english`



List of Assyrian kings | Dynasty of Puzur-Ashur (2025–1749 BC)

King	Reign	Succession
Erishum I e-ri-šu	c. 1905 BC – c. 1867 BC (38 years)	Son of Ilu-shuma
Ikunum I-ku-nu	c. 1867 BC – c. 1860 BC (7 years)	Brother of Erishum I, son of Ilu-shuma
Sargon I Šarru-ukīn "the king is legitimate"	c. 1860 BC – c.1821 BC (39 years)	Son of Ikunum
Puzur-Ashur II Puzur-Aššur "servant of Ashur"	c. 1821 BC – c.1813 BC (8 years)	Son of Sargon I
Naram-Sin Na-ra-am Sîn "beloved of the Moon God Sîn"	c. 1813 BC – c.1769 BC (44 years)	Son of Puzur-Ashur II

Question:
On the Assyrian King List, for whom is Sargon the son and successor of and for whom is he the father and predecessor of?

Step 0: Retrieve original sentence.
On the Assyrian King List, Sargon appears as the son and successor of Ikunum, and the father and predecessor of Puzur-Ashur II.

Step 1: Identify entities to replace: {Puzur-Ashur II)

Knowledge-Based Perturbation

Step 2: Identify columns of entities.
{Entity: Puzur-Ashur II,
Column: King}

Step 3: Select replacement entities from the same column.
{King: Erishum I, Ikunum, Sargon I, Naram-Sin…}

Step 4: Perturb sentence.
On the Assyrian King List, Sargon appears as… predecessor of Erishum I.

Model-Based Perturbation

Step 2: Prepare the context.
On the Assyrian King List, Sargon appears as… predecessor of ______

Step 3: Teacher-force the above context and predict the continuation using a Data-to-Text model.
(**D2T Model** → Sargon II…)

Step 4: Perturb sentence.
On the Assyrian King List, Sargon appears as… predecessor of Sargon II.

Figure 2: Knowledge-based and model-based replacement methods for obtaining unfaithful sentences.

To sample one unfaithful sentence, we select one of these entities, with those located near the end of the sentence to have a higher probability of being selected. This way, for the token-level detection task, the model has more chances of obtaining a reasonable amount of context for performing discrimination. We propose two different methods of determining the candidate replacement, which are illustrated in Figure 2 and described next.

Knowledge-based Method Entities of same types are usually the suspects of the wrong predictions. Identifying such entities in other text generation tasks is nontrivial because their inputs are unstructured texts, while in D2T, the source arranges similar entities together. For example, similar entities can be accessed in the same column for a table input, or they can be obtained by checking their applicability to certain predicates when the input is a semantic triple set. Our knowledge-based replacement method exploits the structure of the input to retrieve similar entities to find replacements that will lead to contradiction. This process is illustrated in Figure 2. Note that for the datasets we experimented on, we assume the entity to be replaced is the only choice to make the sentence entailed, and replacement of any other entities shown in the input table will violate the faithfulness of sentences. This assumption does not necessarily hold in some examples, on which we will elaborate in Section 6.

Model-based Method Another way to obtain replacement candidates is by sampling from a proposal distribution as in ELECTRA. We sample replacements from the baseline Data-to-Text model by teacher forcing partial sentence up to the entity that needs to be replaced. Next we collect the D2T model predictions of the continuation with nucleus sampling (Holtzman et al., 2020), followed by extracting altered entities shown in the predicted continuations to determine the replacement candidates. With this approach, we are able to expose and further train the D2T model with errors of its own predictions.

2.3 Replacement Detection in Generation

For each entailed sentence $Y^{(i)}_{\text{True}}$, we generate $N^{(i)}$ contradictory sentences which we denote as $Y^{(i,j)}_{\text{False}}$ for $j = 1, \ldots, |N^{(i)}|$. The number of contradictory sentences we can generate given an entailed sentence depends on the number of entities found in the original sentence, the input, and the replacement method applied.

As shown in Figure 3, we add a sentence-level replacement detection task to the existing Sequence-to-Sequence framework by eliciting the decoder to generate a probability of the teacher-forced sentence being entailed or contradictory at the end of the generation, similar to the sequence classification usage of BART (Lewis et al., 2020), except that in BART, the same sequence that needs to be classified is fed into both the encoder and decoder. The loss for sentence-level replacement detection is defined by Equation (1).

A more challenging task is to perform a fine-grained, token-level discrimination, as shown in Figure 4. Instead of predicting a discrimination probability at the end of generation, we task the decoder to perform discrimination at every step of token generation. Specifically, we use the per-step last hidden output of the decoder, which encodes the source contexts and teacher-forced partial generation contexts, to compute the discrimination probability with a linear and sigmoid layer. The token-level replacement detection loss is defined by Equation (2).

$$\mathcal{L}_{\mathrm{RD}_{\mathrm{sent}}}(X^{(i)}, Y) = -\left[l \cdot \log p(l|Y, X^{(i)}) + (1-l) \cdot \log\left(1 - p(l|Y, X^{(i)})\right)\right] \quad (1)$$

$$\mathcal{L}_{\mathrm{RD}_{\mathrm{token}}}(X^{(i)}, Y) = -\left[\sum_{t=1}^{|Y|} l_t \cdot \log p(l_t|y_{\le t}, X^{(i)}) + (1-l_t) \cdot \log\left(1 - p(l_t|y_{\le t}, X^{(i)})\right)\right] \quad (2)$$

$$\mathcal{L}_{\mathrm{UL}}(X^{(i)}, Y^{(i,j)}_{\mathrm{False}}) = -\left[\sum_{t=\{t|y_t \in \mathcal{C}^{(i,j)}\}}^{|\mathcal{C}^{(i,j)}|} y_t \cdot \log(1-\hat{y}_t) + (1-y_t) \cdot \log(\hat{y}_t) + \sum_{t=\{t|y_t \notin \mathcal{C}^{(i,j)}\}}^{|Y^{(i,j)}_{\mathrm{False}} \setminus \mathcal{C}^{(i,j)}|} y_t \cdot \log(\hat{y}_t) + (1-y_t) \cdot \log(1-\hat{y}_t)\right] \quad (3)$$

$$\mathcal{L}_{\mathrm{NLL}}(X^{(i)}, Y^{(i)}_{\mathrm{True}}) = -\sum_{t=1}^{|Y^{(i)}_{\mathrm{True}}|} y_t \cdot \log(\hat{y}_t) + (1-y_t) \cdot \log(1-\hat{y}_t) \quad (4)$$

$$\mathcal{L}_{\mathrm{R2D2}} = \sum_{i=1}^{|\mathcal{D}|} \frac{1}{|N^{(i)}|+1}\left[\lambda\left(\mathcal{L}_{\mathrm{NLL}}(X^{(i)}, Y^{(i)}_{\mathrm{True}}) + \sum_{j=1}^{|N^{(i)}|} \mathcal{L}_{\mathrm{UL}}(X^{(i)}, Y^{(i,j)}_{\mathrm{False}})\right) + (1-\lambda)\left(\mathcal{L}_{\mathrm{RD}_{\{\mathrm{sent},\mathrm{token}\}}}(X^{(i)}, Y^{(i)}_{\mathrm{True}}) + \sum_{j=1}^{|N^{(i)}|} \mathcal{L}_{\mathrm{RD}_{\{\mathrm{sent},\mathrm{token}\}}}(X^{(i)}, Y^{(i,j)}_{\mathrm{False}})\right)\right] \quad (5)$$

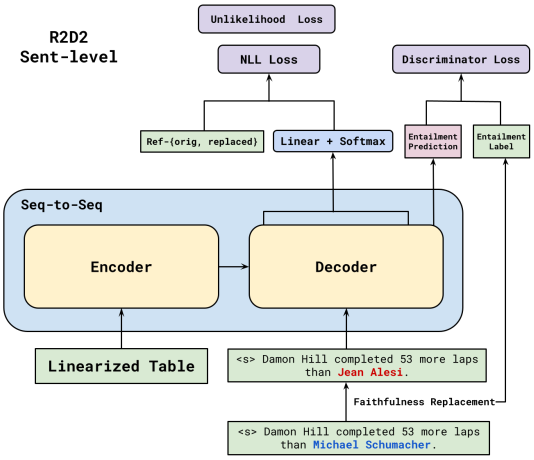


Figure 3: R2D2 sentence-level architecture

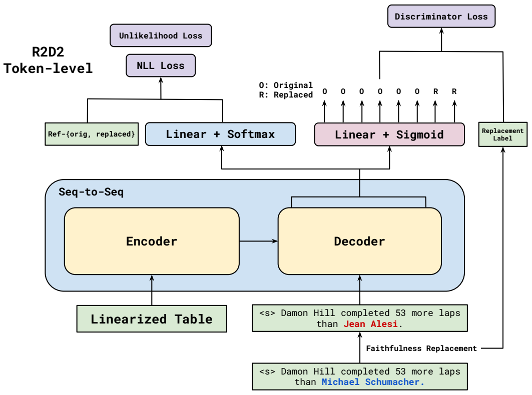


Figure 4: R2D2 token-level architecture

2.4 Replacement Unlikelihood Training

Unlikelihood training is first proposed in (Welleck et al., 2019) to address the repetition problem of the neural text generation. This training objective aims to decrease the decoder's probability of generating tokens that are already seen in the teacher-forced generation contexts. The applicability of this objective to the Data-to-Text task is also argued in (Uehara et al., 2020).

Instead of using the generated tokens to construct the negative candidate set defined for each step, we define the sentence-level negative candidate span $\mathcal{C}^{(i,j)}$ for each contradictory sentence $Y^{(i,j)}_{\mathrm{False}}$. The span contains, for each time step, one replaced token that should have a low probability of being generated. We calculate the sequence-level unlikelihood loss for this replaced token span and apply regular likelihood loss for other original tokens, which we denote as $Y^{(i,j)}_{\mathrm{False}} \setminus \mathcal{C}^{(i,j)}$. We denote the per step prediction as $\hat{y}_t = p(y_t|y_{<t}, X^{(i)})$. The unlikelihood loss for the entire sentence is specified in Equation (3).

2.5 R2D2 Fine-tuning

We propose the final R2D2 fine-tuning loss objective as in equation (5). It combines a generation task loss component and a discrimination task loss component. Each of them is calculated from one entailed instance and $N^{(i)}$ contradictory instances. The generation task component consists of a regular negative log likelihood loss for the entailed instance, as described in Equation (4), and an unlikelihood loss for the contradictory instances. The discrimination task component contains either sentence-level or token-level replacement detection loss for both entailed and contradictory instances. We use λ to balance the importance between the two loss components.

3 Experiments

We first introduce the datasets that we experiment with in Section 3.1, and the metrics we adopted for evaluation in Section 3.2. Then we report the baseline models we are comparing with, and the implementation and training details in Section 3.3. In Section 3.4 and 3.5, we report and analyze both the automatic evaluation results and the human evaluation results.

3.1 Datasets

`FeTaQA` (Nan et al., 2022) is a free-form table question answering dataset. It introduces a task that requires retrieving the correct contents from the table based on the question, integrating and inferring from the retrieved facts, and generating a free-form answer. Sentences that contain erroneous selections of facts, even if they appear in the input table, are still considered as unfaithful, for being inconsistent with the input question.

`LogicNLG` (Chen et al., 2020a) is a table-to-text dataset that requires generation of logically entailed sentences, with no indication of what is considered as salient given a table. Since there are numerous entailed facts that are different from the references in the surface-form, they propose input-based metrics that compare the facts in the generated sentence and those in the input. It is worth noting that faithfulness to the input is more important for LogicNLG then faithfulness to the references.

`ToTTo` (Parikh et al., 2020) is a table-to-text dataset that contains annotations of salient content of tables (highlighted table cells). The task does not require any content selection (when only highlighted cells constitute the input), but only text planning and surface realization of the inputs, which are expected to be described with full coverage.

3.2 Evaluation Metrics

We report results of a variety of automatic evaluation metrics used in the past studies to provide a comprehensive comparison of existing methods and our proposed method. We include fact-verification based metrics, NLI-Acc (Chen et al., 2020a,b), which specifically aims to evaluate the faithfulness of the sentences. We also report string-based metrics that evaluate the string match between predictions and references, such as sacre-BLEU (Post, 2018), ROUGE-{1, 2, L} (Lin, 2004), TER (Snover et al., 2006), METEOR (Banerjee and Lavie, 2005), PARENT (Dhingra et al., 2019) (which also leverages the input data).

NE-based Evaluation Metrics To better understand how the D2T system's retrieval capability correlates with faithfulness, we propose information-extraction based metrics that compare the named entities contained in the generated sentences to those contained in the reference sentences or input data. We believe these metrics help us better distinguish between the unfaithfulness caused by wrong retrieval of entities and that caused by wrong prediction of predicates/relations. Specifically, we propose the following indicators:

- **Reference coverage (RC)**: percentage of entities in the reference that are also shown in the prediction.
- **Ref-hit & Input-hit (RI)**: percentage of entities shown in the prediction that are shown in both the reference and input table.
- **Ref-hit & Input-miss (RM)**: percentage of predicted entities that are shown in the reference but not the input table. This case is rare since it indicates the existence of entities that are not input-grounded in the reference.
- **Ref-miss & Input-hit (MI)**: percentage of predicted entities that are shown in the table, but not in the reference. This case identifies wrong or unnecessary retrieval of entities from the input (when the indication of saliency is evident).
- **Ref-miss & Input-miss (MM)**: percentage of predicated entities that are neither shown in the table nor the reference. This case identifies the prediction of entities likely by hallucination.

Figure 5 and Figure 6 in the Appendix show the correlation between NE-based metrics and sacre-BLEU, NLI-Acc, respectively. As expected, the reference coverage rate positively correlates with both metrics. While both reference hit and input hit are important, the rate of predicted entities not shown in reference negatively correlates with sacre-BLEU (MI and MM) and NLI-Acc (MI). The trend is less clear for RM and MM since these are rare cases, which can also be shown in Table 4. In Section A.1 of the Appendix, we also test how the

Systems	Fact-based	String-based				
	NLI-Acc	sacreBLEU	Rouge-1/2/L	PARENT	TER	METEOR
Xie et al. (2022)	-	29.9	61.77/39.44/51.93	-	-	48.53
Reg-FT	74.79	30.5	63.47/41.77/54.04	44.24	66.37	55.2
R2D2-FT	**77.93**	**31.5**	**63.50/41.71/54.05**	**45.32**	68.55	**56.27**

Table 1: Automatic evaluation results of different systems on FeTaQA test split.

Systems	Fact-based		String-based					
	NLI-Acc	SP-Acc	BLEU-1/2/3	sacreBLEU	Rouge-1/2/L	PARENT	TER	METEOR
Chen et al. (2021b)	76.9	43.9	49.5/28.6/15.3	-	-	-	-	-
Reg-FT	84.11	45.97	51.63/32.24/18.75	18.2	42.74/20.89/36.77	32.36	86.38	36.55
R2D2-FT	**85.57**	**50.80**	**51.76/32.42/18.65**	**18.5**	42.63/20.73/36.84	31.38	**80.97**	35.73

Table 2: Automatic evaluation results of different systems on LogicNLG test split.

automatic metrics we reported could reliably detect the unfaithfulness of the sentences.

3.3 Experiment Settings

Baselines The state-of-the-art system for the Data-to-Text task is fine-tuned T5 model (Raffel et al., 2020). We fine-tune T5 ourselves and report evaluations on FeTaQA, LogicNLG and ToTTo, so that the learning objective is the key control variable in our comparison.

Implementations We use T5-base as the pre-trained checkpoint from which we fine-tune either regularly (Reg-FT) or using our proposed method (R2D2-FT). For R2D2-FT, we initialize our model from a checkpoint that has been fine-tuned regularly for 15 epochs, and train the additional linear layer for sentence or token replacement detection from random initialization. We find this fine-tuning warmup help improve the performance in general. We use the Adafactor optimizer (Shazeer and Stern, 2018) with a learning rate of 5e-5. We use the batch size of 8 for FeTaQA and 32 for the others. Our models are trained on one NVIDIA GeForce RTX 3090 GPU, and each experiment takes around 5-20 hours depending on the dataset size.

R2D2 Configuration To assess the necessity of the discrimination loss and unlikelihood loss, we experimented fine-tuning T5 only with the discrimination loss, or only with the unlikelihood loss, or both (all in addition to the NLL loss). For discrimination loss, we also experiment with adding sentence-level or token-level discrimination to investigate the effect of discrimination granularity in assisting faithful text generation. For all the training variants above, we also compare two methods of obtaining the contradictory sentences, knowledge-based and model-based methods. Since the number of contradictory sentences obtained (which we denote as $N^{(i)}$ in Section 2.5) varies depending on the method used (as shown in Table 7 of the Appendix), we also experiment with using different numbers of contradictory sentences in the R2D2 fine-tuning: $N^{(i)} = 1$ (xsmall), 3 (small), 5 (medium), 10 (large) or max (full). Since the maximum size of the perturbations obtained by model-based method is small, we only compared xsmall and full for the model-based method setting.

3.4 Automatic Evaluation

We report the performance of the previous state-of-the-art system, T5 fine-tuned only with negative log likelihood (NLL) loss by ourselves, and the best T5 fine-tuned with R2D2 loss for FeTaQA (Table 1), LogicNLG (Table 2) and ToTTo (Table 3), based on metrics used in the existing literature. In Table 4, we also report their performances using the NE-based metrics that we proposed. We also report the full experiment results of FeTaQA that contain evaluations of different R2D2 configurations in Table 8 of the Appendix.

We obverse that across all the datasets, most of the systems fine-tuned with different R2D2 configurations are able to perform better than system that is fine-tuned only with NLL loss. As expected, the improvements are more evident in the fact-verification-based metrics that evaluate the faithfulness of the sentences. As shown in Table 8, we find that the best R2D2 configuration requires both

Systems	Split	Fact-based	String-based				
		NLI-Acc	sacreBLEU	Rouge-1/2/L	PARENT	TER	METEOR
Kale and Rastogi (2020)	All	-	47.7	-	57.1	-	-
	Overlap	-	-	-	-	-	-
	Nonoverlap	-	39.6	-	52.6	-	-
Reg-FT	All	90.29	48.7	75.92/55.99/67.40	58.43	48.53	71.01
	Overlap	91.17	56.6	79.95/62.39/72.78	62.88	40.85	75.56
	Nonoverlap	89.42	41.0	72.04/49.80/62.20	54.14	55.69	66.54
R2D2-FT	All	**91.27**	**49.2**	75.54/55.59/67.06	**59.05**	50.40	**71.99**
	Overlap	**91.70**	**56.7**	79.42/61.68/72.14	**62.89**	42.72	**76.47**
	Nonoverlap	**90.86**	**41.9**	71.79/49.69/62.15	**55.35**	57.57	**67.67**

Table 3: Automatic evaluation results of systems on ToTTo development split.

the discrimination and the unlikelihood learning objectives with $\lambda = 0.5$ (finding from a parameter sweep of 0.2, 0.5 and 0.8). The contradictory sentences obtained by knowledge-based perturbation are more beneficial than those obtained by model-based perturbation. We also find that the granularity (sentence/token-level) of the discrimination loss does not seem to affect the performance much, and that the system performance does not necessarily improve as we increase the number of unfaithful sentences used for fine-tuning, and that the best configuration for $N^{(i)}$ seems to be 3-5 in most cases.

We examine the improvement of faithfulness using the NE-based metrics, and find that the coverage of the entities appeared in the reference (RC) improves across all datasets. For FeTaQA, we notice that our system is able to retrieve *input-grounded entities* more accurately (shown by increased RI and decreased MI scores). For LogicNLG which has no right or wrong retrieval of input-grounded entities as long as the description of them is faithful, we obverse an evident decline of MM and increments of both RI and MI (with more evident gain in MI), indicating that our system is able to reduce hallucinations of irrelevant entities and instead retrieving input-grounded ones.

Dataset	Systems	Split	RC	RI	RM	MI	MM
FeTaQA	**Reg**	A	72.30	69.88	1.48	24.90	3.73
	R2D2	A	**73.06**	**70.92**	**1.62**	**23.82**	**3.64**
LogicNLG	**Reg**	A	37.29	26.07	0.97	61.60	11.36
	R2D2	A	**37.93**	**26.62**	**0.44**	**67.12**	**5.82**
ToTTo	**Reg**	A	82.47	74.95	3.14	17.66	4.24
		O	84.87	77.95	3.40	15.10	3.54
		N	80.14	72.04	2.89	20.14	4.92
	R2D2	A	**83.24**	74.06	3.26	18.31	4.37
		O	**85.45**	76.90	3.63	15.70	3.77
		N	**81.08**	71.30	2.89	20.85	4.95

Table 4: NE-based automatic evaluation result. A, O, N stands for All, Overlap and Nonoverlap.

Dataset	Systems	Split	Faithfulness Agreement / κ	Coverage w.r.t Ref Agreement / κ
FeTaQA	**Reg-FT**	A	61.33 / 0.62	54.83 / 0.46
	R2D2-FT	A	**68.67** / 0.61	**61.83** / 0.51
LogicNLG	**Reg-FT**	A	40.67 / 0.71	69.00 / 0.36
	R2D2-FT	A	**41.17** / 0.74	66.50 / 0.44
ToTTo	**Reg-FT**	A	81.00 / 0.41	81.17 / 0.37
		O	83.66 / 0.51	82.66 / 0.38
		N	78.32 / 0.33	79.66 / 0.37
	R2D2–FT	A	**83.16** / 0.37	**84.67** / 0.34
		O	**84.34** / 0.45	**88.66** / 0.25
		N	**82.00** / 0.31	**80.66** / 0.40

Table 5: Human evaluation result. A, O, N stands for All, Overlap and Nonoverlap. We measure the total agreement in % and the Fleiss' Kappa (κ) (Fleiss, 1971).

3.5 Human Evaluation

Since the automatic evaluations are not always reliable in determining the faithful aspect of a sentence, which can be seen in our metrics reliability test shown in Table 6: around 14% faithful sentences are deemed to be unfaithful by the NLI-Acc metric, and more importantly, it fails to identify around 36% of the unfaithful sentences. We conduct the human evaluation based on two criteria: a sentence is (1) *faithful* if **all facts** contained are entailed by the input, and when a question is present in the input, the sentence only contains **necessary facts**; (2) *adequate with respect to reference* if the sentence contains **same or more facts** than the reference. We asked three human evaluators to evaluate 200 samples of each dataset (100 samples in each of the overlap/nonoverlap split for ToTTo), and each sample is provided with all the inputs, the reference, and two system generated sentences. We report the percentage of faithful and adequate sentences generated by the baseline system and our system on all datasets in Table 5, and the results validate R2D2's effectiveness in faithful text generation. We notice that on LogicNLG, the R2D2 generated sentences' coverage with respect to the reference is lower than that of the baseline model's generations, we suspect that this is because the R2D2 generated sentences may contain facts that are different from the facts shown in the reference.

4 Related Work

4.1 Unfaithfulness in Text Generation

In the context of text generation, hallucination refers to the phenomenon of neural models "generating unfaithful or nonsensical text" (Ji et al., 2022). Reasons for such hallucinations are poor data collection, training design (such as the exposure bias), or that the task expects more output diversity. Metrics based on information extraction, question answering, and natural language inference, have been proposed to measure such hallucination, which we employ to evaluate the performance and faithfulness of R2D2.

4.2 Contrastive Learning

Contrastive learning (Hadsell et al., 2006) tasks the model with maximizing the representation similarity between neighboring examples while minimizing the similarity between distant examples. Contrastive learning has recently been used in various NLP tasks, including language modeling (Arora et al., 2022), machine translation (Yang et al., 2019; Pan et al., 2021), anomaly detection (Manolache et al., 2021), commonsense reasoning (Zhou et al., 2021), text summarization (Cao and Wang, 2021; Liu and Liu, 2021; Xu et al., 2021; Sun and Li, 2021; Wang et al., 2021; Liu et al., 2022), and data-to-text generation (Uehara et al., 2020). Unlike Uehara et al. (2020), in which unfaithful sentences are obtained by replacing a set of keywords (such as replacing *low* to *high*, *gain* to *drop*) that only apply to the finance domain, we propose domain-independent methods for sampling unfaithful sentences either by exploiting the structure of input knowledge or utilizing the D2T model's own mistakes.

4.3 Unlikelihood Training

To address the degeneration problems of models trained only with Maximum Likelihood Estimation, many works have proposed alternative approaches (Tu et al., 2016; Li et al., 2020; Holtzman et al., 2020; Lin et al., 2021). Among them, unlikelihood training was introduced as a means of decreasing the probability that the model generates certain tokens (Welleck et al., 2019). In a D2T context, we adopt unlikelihood training to decrease the probability that the model generates tokens which are not entailed by the given contexts.

4.4 Evaluation Metrics

Ideally, a data-to-text model should be evaluated based on its ability to generate logical sentences verified by the provided reference data. Current methods however, typically only compare the model output summary to the gold summary. This includes n-gram based (e.g. BLEU, ROUGE, and METEOR) or edit distance based metrics (e.g. TER) (Sai et al., 2022), or embedding-based similarity metrics (e.g. BERTScore) (Zhang et al., 2019). Another set of metrics compare the information present in the output and the label. This is done by extracting subject, object, and their relations in the output and label, and comparing both sets of elements (Wiseman et al., 2017). We evaluate our model using multiple metrics to understand different aspects of its performance. A comprehensive investigation of the current evaluation practices for NLG tasks can be found in Gehrmann et al., 2022.

4.5 Natural Language Inference

Natural language inference (NLI) refers to the task of classifying whether a hypothesis entails, contradicts, or is unrelated to a premise (Bowman et al., 2015). In the context of D2T, NLI can be used to evaluate whether a model's generated text can be inferred from the input table (Chen et al., 2020a). In line with the work of TabFact (Chen et al., 2020b), LogicNLG (Chen et al., 2020a), and SnowBall (Shu et al., 2021), R2D2 incorporates this idea into data-to-text training by using NLI as a learning objective during the training procedure.

5 Conclusion

In this work, we introduced R2D2, a training framework that mitigates the unfaithful text generation problem for the D2T task. Training with the regular maximum likelihood loss can lead to generation of sentences that are similar to the references but are unfaithful to the input. We therefore propose to add a discrimination task and an unlikelihood training to encourage the model to generate separable representations of these critical sentences. We proposed two methods of sampling these unfaithful sentences: the knowledge-based method exploits the structure of the input knowledge, and the model-based method samples the D2T model's own mistakes. We proposed NE-based metrics that assess the entity retrieval capability of the Data-to-Text systems, as we argued the incompetence of which is one of the leading causes of unfaithfulness. We experimented on multiple Data-to-Text datasets of different task constructs, and achieved noticeable improvements over the state-of-the-art performance.

6 Limitations

There are some limitations of our knowledge-based method for obtaining the contradictory sentences, as its validity depends on the type of sentences observed in the data-to-text datasets. Comparing the effectiveness of R2D2 on different datasets and with different evaluations, we found that FeTaQA and ToTTo benefit more than LogicNLG. We speculate this is because many sentences of LogicNLG describe some entailed facts of entities of a single table column (usually involving comparisons), which usually contain single and less restricted predicate that could be applied to many homogeneous entities, and this would invalidate our perturbation methods. We provide one such example in Figure 7 of the Appendix. We also notice that for ToTTo, the improvement is less evident than that for FeTaQA. Besides less room for improvement, we observe no evident change in the entities retrieved by both systems compared with those in the reference or input, while human evaluation indicates there are still around 17% unfaithful sentences. We speculate the source of unfaithfulness of these sentences are due to wrong predictions of relations/predicates, which are not captured and included into the R2D2 fine-tuning by our current perturbation method. To avoid invalidation of perturbation (as in the case of LogicNLG) and also to capture erroneous relation predictions, a better perturbation method has to operate on fact triples instead of entities, but this requires a reliable and domain-independent fact extraction system, which we will explore in future.

References

Kushal Arora, Kurt Shuster, Sainbayar Sukhbaatar, and Jason Weston. 2022. Director: Generator-classifiers for supervised language modeling.

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.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference.

Shuyang Cao and Lu Wang. 2021. CLIFF: Contrastive learning for improving faithfulness and factuality in abstractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6633–6649, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2017. Faithful to the original: Fact aware neural abstractive summarization.

Sihao Chen, Fan Zhang, Kazoo Sone, and Dan Roth. 2021a. Improving faithfulness in abstractive summarization with contrast candidate generation and selection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5935–5941, Online. Association for Computational Linguistics.

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020a. Logical natural language generation from open-domain tables. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7929–7942, Online. Association for Computational Linguistics.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020b. Tabfact : A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia.

Wenqing Chen, Jidong Tian, Yitian Li, Hao He, and Yaohui Jin. 2021b. De-confounded variational encoder-decoder for logical table-to-text generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5532–5542, Online. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William Cohen. 2019. Handling divergent reference texts when evaluating table-to-text generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4884–4895, Florence, Italy. Association for Computational Linguistics.

J. L. Fleiss. 1971. Measuring nominal scale agreement among many raters. 76(5):378–382.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostela, Spain. Association for Computational Linguistics.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *J. Artif. Int. Res.*, 61(1):65–170.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2022. Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text.

Tanya Goyal and Greg Durrett. 2020. Evaluating factuality in generation with dependency-level entailment. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3592–3603, Online. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2021. Annotating and modeling fine-grained factuality in summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1449–1462, Online. Association for Computational Linguistics.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, page 1735–1742, USA. IEEE Computer Society.

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

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2022. Survey of hallucination in natural language generation.

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

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Margaret Li, Stephen Roller, Ilia Kulikov, Sean Welleck, Y-Lan Boureau, Kyunghyun Cho, and Jason Weston. 2020. Don't say that! making inconsistent dialogue unlikely with unlikelihood training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4715–4728, 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.

Xiang Lin, Simeng Han, and Shafiq Joty. 2021. Straight to the gradient: Learning to use novel tokens for neural text generation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6642–6653. PMLR.

Yixin Liu and Pengfei Liu. 2021. SimCLS: A simple framework for contrastive learning of abstractive summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1065–1072, Online. Association for Computational Linguistics.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. BRIO: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2890–2903, Dublin, Ireland. Association for Computational Linguistics.

Andrei Manolache, Florin Brad, and Elena Burceanu. 2021. DATE: Detecting anomalies in text via self-supervision of transformers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 267–277, Online. Association for Computational Linguistics.

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

Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Nick Schoelkopf, Riley Kong, Xiangru Tang, Mutethia Mutuma, Ben Rosand, Isabel Trindade, Renusree Bandaru, Jacob Cunningham, Caiming Xiong, and Dragomir Radev. 2022. FeTaQA: Free-form Table Question Answering. *Transactions of the Association for Computational Linguistics*, 10:35–49.

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.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 244–258, Online. 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. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186, Online. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. 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. *Journal of Machine Learning Research*, 21(140):1–67.

Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Studies in Natural Language Processing. Cambridge University Press.

Ananya B. Sai, Akash Kumar Mohankumar, and Mitesh M. Khapra. 2022. A survey of evaluation metrics used for nlg systems. *ACM Comput. Surv.*, 55(2).

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR.

Chang Shu, Yusen Zhang, Xiangyu Dong, Peng Shi, Tao Yu, and Rui Zhang. 2021. Logic-consistency text generation from semantic parses. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4414–4426, Online. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Shichao Sun and Wenjie Li. 2021. Alleviating exposure bias via contrastive learning for abstractive text summarization. *CoRR*, abs/2108.11846.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany. Association for Computational Linguistics.

Yui Uehara, Tatsuya Ishigaki, Kasumi Aoki, Hiroshi Noji, Keiichi Goshima, Ichiro Kobayashi, Hiroya Takamura, and Yusuke Miyao. 2020. Learning with contrastive examples for data-to-text generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2352–2362, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Danqing Wang, Jiaze Chen, Hao Zhou, Xipeng Qiu, and Lei Li. 2021. Contrastive aligned joint learning for multilingual summarization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2739–2750, Online. Association for Computational Linguistics.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. *arXiv preprint arXiv:2201.05966*.

Shusheng Xu, Xingxing Zhang, Yi Wu, and Furu Wei. 2021. Sequence level contrastive learning for text summarization. *CoRR*, abs/2109.03481.

Zonghan Yang, Yong Cheng, Yang Liu, and Maosong Sun. 2019. Reducing word omission errors in neural machine translation: A contrastive learning approach. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6191–6196, Florence, Italy. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert.

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*.

Wangchunshu Zhou, Dong-Ho Lee, Ravi Kiran Selvam, Seyeon Lee, and Xiang Ren. 2021. Pre-training text-to-text transformers for concept-centric common sense. In *International Conference on Learning Representations*.

A Appendix

A.1 Evaluation Metric Reliability Test

Since many existing automatic metrics for text generation tasks are not proposed with an aim of reflecting the faithfulness of the sentences, an examination of all metrics reported in our work is crucial for interpreting the results. As we are able to reliably generate an unfaithful version of most of the reference texts, we contaminate the references in the FeTaQA test split in a controlled manner: we generate five variants of texts with different percentage of the references being replaced with their unfaithful parallel (0% version contains only the references and 100% version contains only the unfaithful sentences). We evaluate the variants that are contaminated to different degrees using the evaluation metrics we reported, in order to investigate how reliable they are in reflecting the faithfulness of any system generated sentences.

As shown in Table 6, most of the metrics are able to reflect the degree of unfaithfulness contained in the prediction texts, though our test only contains the type of unfaithfulness that originates from erroneous selection of entities. A more rigorous study would test other types of unfaithfulness, such as wrong prediction of relations or arrangement of entities. Nevertheless, we observe that some metrics, especially NLI-Acc, are more sensitive to the type of unfaithfulness that we tested, while an unfaithful sentence can still obtain a very high BERTScore (Zhang et al., 2020).

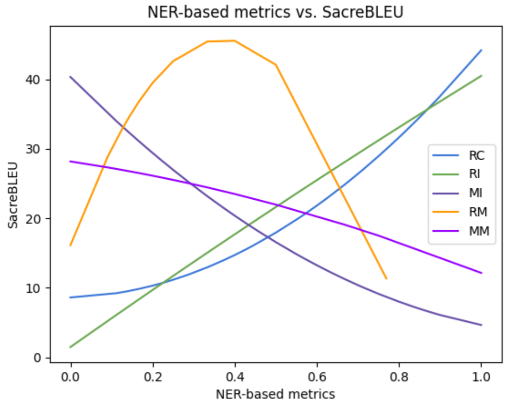


Figure 5: NE-based metrics vs sacreBLEU

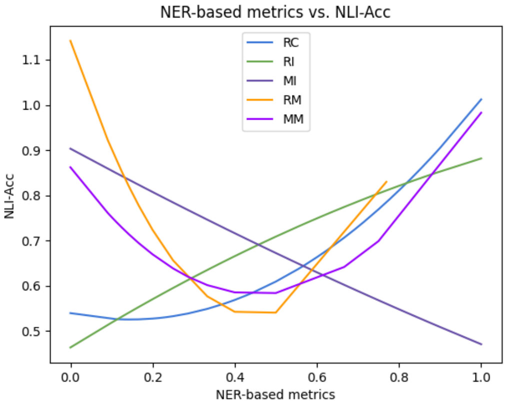


Figure 6: NE-based metrics vs NLI-Acc

% Unfaithful Sentences	Fact-based	String-based						NE-based				
	NLI-Acc	sacreBLEU	Rouge-1/2/L	PARENT	TER	METEOR	BERTScore	RC	RI	RM	MI	MM
0% (ref.)	86.17	100.0	100/100/100	89.87	0.0	99.99	100.0	99.95	94.15	5.80	0.0	0.05
25%	73.29	96.1	97.62/96.31/97.62	87.23	2.57	98.08	99.43	96.37	90.82	5.67	2.98	0.53
50%	61.21	92.2	95.28/92.63/95.29	84.63	5.18	96.24	98.87	93.01	87.53	5.55	5.86	1.06
75%	48.63	88.3	92.87/88.87/92.87	81.91	7.84	94.36	98.29	89.32	84.33	5.31	8.81	1.54
100%	36.20	84.4	90.47/85.13/90.48	79.16	10.36	92.48	97.72	85.70	80.93	5.24	11.72	2.11

Table 6: Reliability test result of evaluation metrics for detecting unfaithfulness

Dataset	Original Train-split Size	Knowledge-based	Model-based
FeTaQA	7,325	238,891 (×32.6)	25,313 (×3.5)
LogicNLG	21,873	413,247 (×18.9)	91,183 (×4.2)
ToTTo	120,761	1,165,067 (×9.6)	450,770 (×3.7)

Table 7: Number of obtainable contradictory sentences for the train-split of all datasets.

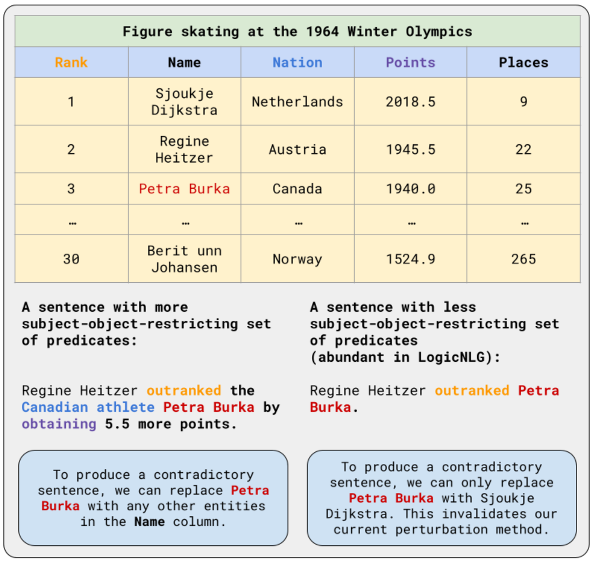


Figure skating at the 1964 Winter Olympics

Rank	Name	Nation	Points	Places
1	Sjoukje Dijkstra	Netherlands	2018.5	9
2	Regine Heitzer	Austria	1945.5	22
3	Petra Burka	Canada	1940.0	25
…	…	…	…	…
30	Berit unn Johansen	Norway	1524.9	265

A sentence with more subject-object-restricting set of predicates:

Regine Heitzer outranked **the** Canadian athlete Petra Burka **by** obtaining **5.5 more points.**

To produce a contradictory sentence, we can replace Petra Burka with any other entities in the **Name** column.

A sentence with less subject-object-restricting set of predicates (abundant in LogicNLG):

Regine Heitzer outranked Petra Burka.

To produce a contradictory sentence, we can only replace Petra Burka with Sjoukje Dijkstra. This invalidates our current perturbation method.

Figure 7: Sentences for which our current perturbation methods do not apply.

Discrimination Loss	**Discrimination Granularity**	**Unlikelihood Loss**	**Perturbation Method**	**Perturbation Size**	**Fact-based**	**String-based**				
					NLI-Acc	**sacreBLEU**	**Rouge-1/2/L**	**PARENT**	**TER**	**METEOR**
✓	**sent-level**	✗	**Model based**	`xsmall`	76.83	30.5	63.46/41.57/53.84	44.49	66.47	55.33
				`full`	74.69	29.2	63.08/41.03/53.57	43.01	66.76	54.39
✓	**sent-level**	✗	**Knowledge based**	`xsmall`	75.79	29.7	63.26/41.08/53.54	43.21	66.65	54.69
				`small`	74.59	29.7	63.53/41.54/54.06	43.43	65.75	54.67
				`medium`	77.13	30.2	63.47/41.43/53.68	43.99	66.96	55.09
				`large`	74.64	29.2	62.57/40.74/53.22	42.74	66.93	53.76
				`full`	73.99	29.4	63.26/41.24153.83	43.45	66.58	54.32
✓	**token-level**	✗	**Model based**	`xsmall`	75.39	30.3	63.27/40.96/53.66	43.39	67.40	55.16
				`full`	74.44	30.4	63.34/41.24/53.92	44.11	67.30	55.58
✓	**token-level**	✗	**Knowledge based**	`xsmall`	74.64	30.0	63.01/40.38/53.16	43.15	68.68	55.46
				`small`	75.74	30.8	63.98/41.92/54.39	44.35	66.81	55.75
				`medium`	75.54	30.5	63.41/41.42/53.93	44.36	67.47	55.60
				`large`	74.14	29.9	63.41/41.16/53.68	43.66	67.61	55.31
				`full`	74.94	29.2	63.23/40.73/53.64	42.62	67.89	54.33
✗	**-**	✓	**Model based**	`xsmall`	74.34	30.2	63.06/40.56/53.41	43.62	68.65	55.22
				`full`	77.33	29.9	62.27/40.16/52.50	43.00	73.97	54.95
✗	**-**	✓	**Knowledge based**	`xsmall`	75.39	30.0	63.00/40.41/53.49	42.85	67.95	54.93
				`small`	77.93	31.1	63.74/41.68/54.26	44.59	68.43	56.11
				`medium`	77.58	30.8	63.34/41.08/53.70	43.87	68.56	55.77
				`large`	76.29	31.5	63.11/41.20/53.82	45.11	69.93	56.30
				`full`	77.68	31.0	63.17/40.96/53.53	44.20	70.97	56.24
✓	**sent-level**	✓	**Model based**	`xsmall`	77.58	30.7	63.30/41.14/53.34	44.18	68.48	55.47
				`full`	75.39	29.6	62.43/40.33/52.95	42.45	69.75	54.26
✓	**sent-level**	✓	**Knowledge based**	`xsmall`	76.14	30.4	63.36/41.03/53.64	43.61	67.89	55.61
				`small`	76.44	31.4	63.56/41.33/53.93	44.68	68.95	56.34
				`medium`	76.59	30.7	63.11/40.75/53.58	43.83	69.73	55.86
				`large`	78.33	31.2	63.41/41.21/53.87	44.87	69.37	56.57
				`full`	78.18	30.9	62.95/40.95/53.26	44.45	71.78	55.81
✓	**token-level**	✓	**Model based**	`xsmall`	75.28	29.8	62.82/40.33/53.10	42.72	68.92	54.48
				`full`	75.19	29.2	61.88/39.49/52.11	41.68	72.00	54.05
✓	**token-level**	✓	**Knowledge based**	`xsmall`	75.79	30.7	63.52/41.02/53.75	43.83	68.09	56.19
				`small`	77.73	31.2	63.30/41.01/53.54	44.55	69.52	56.26
				`medium`	77.93	31.5	63.50/41.71/54.05	45.32	68.55	56.27
				`large`	76.78	30.9	62.67/40.76/53.41	43.93	71.47	55.59
				`full`	78.28	30.9	62.50/40.43/53.07	44.29	71.24	55.69

Table 8: Full `R2D2` experiment results for FeTaQA.