applied sciences

MDPI

Article

Quality Control for Distantly-Supervised Data-to-Text Generation via Meta Learning

Heng Gong [1], Xiaocheng Feng [1,2] and Bing Qin [1,2,*]

[1] Harbin Institute of Technology, Harbin 150001, China
[2] Peng Cheng Laboratory, Shenzhen 518000, China
* Correspondence: qinb@ir.hit.edu.cn

Abstract: Data-to-text generation plays an important role in natural language processing by processing structured data and helping people understand those data by generating user-friendly descriptive text. It can be applied to news generation, financial report generation, customer service, etc. However, in practice, it needs to adapt to different domains that may lack an annotated training corpus. To alleviate this dataset scarcity problem, distantly-supervised data-to-text generation has emerged, which constructs a training corpus automatically and is more practical to apply to new domains when well-aligned data is expensive to obtain. However, this distant supervision method of training induces an over-generation problem since the automatically aligned text includes hallucination. These expressions cannot be inferred from the data, misguiding the model to produce unfaithful text. To exploit the noisy dataset while maintaining faithfulness, we empower the neural data-to-text model by dynamically increasing the weights of those well-aligned training instances and reducing the weights of the low-quality ones via meta learning. To our best knowledge, we are the first to alleviate the noise in distantly-supervised data-to-text generation via meta learning. In addition, we rewrite those low-quality texts to provide better training instances. Finally, we construct a new distantly-supervised dataset, DIST-ToTTo (abbreviation for Distantly-supervised Table-To-Text), and conduct experiments on both the benchmark WITA (abbreviation for the data source Wikipedia and Wikidata) and DIST-ToTTo datasets. The evaluation results show that our model can improve the state-of-the-art DSG (abbreviation for Distant Supervision Generation) model across all automatic evaluation metrics, with an improvement of 3.72% on the WITA dataset and 3.82% on the DIST-ToTTo dataset in terms of the widely used metric BLEU (abbreviation for BiLingual Evaluation Understudy). Furthermore, based on human evaluation, our model can generate more grammatically correct and more faithful text compared to the state-of-the-art DSG model.

Keywords: data-to-text generation; Natural Language Generation; natural language processing; deep learning; meta learning; Artificial Intelligence

Citation: Gong, H.; Feng, X.; Qin, B. Quality Control for Distantly-Supervised Data-to-Text Generation via Meta Learning. *Appl. Sci.* **2023**, *13*, 5573. https://doi.org/10.3390/app13095573

Academic Editor: Yu-Dong Zhang

Received: 4 April 2023
Revised: 19 April 2023
Accepted: 26 April 2023
Published: 30 April 2023

1. Introduction

With much structural data in our life [1–4], data-to-text generation has become an important text generation task in the field of natural language processing that can help people better understand the meaning behind those data [5]. For example, given statistics of basketball games [2], data in the stock market [4] or structural data in the knowledge base [6], a data-to-text generation system can automatically generate a corresponding text draft for the report, boosting the efficiency of related business. It has many applications, such as news generation, financial report generation, customer service, etc. Take the first row in Figure 1 as an example. Given the triples from the knowledge base (e.g., <Albania–Japan relations, instance_of, bilateral relation>), the model is expected to reflect information in those triples with user-friendly text.

This task has recently attracted much attention in natural language processing, and multiple well-aligned datasets [1–3,7,8] have been proposed. However, those datasets are

confined to specific domains or data sources as it is difficult and expensive to construct these datasets, thus, making it difficult to quickly adapt to different domains.

Quality	Data	Text
🙂	<Albania – Japan relations, instance_of, bilateral relation>	Albania – Japan relations are the bilateral relations between Albania and Japan .
😐	<Johnny Apollo, director, Henry Hathaway> <Johnny Apollo, genre, crime film>	Johnny Apollo is a *1940* crime film directed by Henry Hathaway.
😟	<Kirsten Peetoom, date_of_birth , 1 January 1988>	Kirsten Peetoom (born 1 *October* 1988) is a *Dutch professional racing cyclist*.

Figure 1. Three training instances in a distantly-supervised data-to-text generation dataset, consisting of structured data and its corresponding text. Hallucination/incorrect text (i.e., noise) are in red and italics, respectively. Quality indicates our view on the quality of individual training instances and is not present in the dataset.

Traditionally, the data-to-text generation task has two major types of methods: pipeline-style systems [9,10] and end-to-end models [2,3,11]. The former breaks the generation process into multiple stages that decide what information in the data to describe and then to transform that information into text. In recent years, with the help of high-quality datasets [2,3] and the success of deep learning, end-to-end models [12–14], which take the structured data as input and directly generate the text through neural language models, have become mainstream.

Many researchers have explored improving end-to-end models from different angles. Some researchers have focused on improving the content selection ability [3,12] so that the model can generate more informative text. Some researchers explored enhancing the end-to-end model's ability to generate more faithful text [13,14], that is, the information in the text is more consistent with the data. To increase the informativeness of the text, rather than simply describe the information in the data, some researchers [15–19] explored incorporating reasoning ability into the end-to-end model so that they could provide more insights about the data.

Furthermore, some researchers explored few-shot data-to-text generation [20–23], which trains the end-to-end model with a limited training corpus. However, the studies above require well-aligned training datasets in order to train the data-driven end-to-end model. However, in practice, the model needs to adapt to different domains that may lack an annotated training corpus. In order to alleviate this kind of data scarcity problem, a new task, called distantly-supervised data-to-text generation [6], was proposed that automatically constructs a corpus for data-to-text generation without well-aligned data at present.

This task extracts triples from text to form the dataset. However, due to the imperfect information-extraction technique, texts in the dataset contain hallucinations that cannot be inferred from the data. These noises can misguide the model to produce unfaithful text [24] and make it difficult for the previous neural data-to-text generation models that are not designed to adapt to this scenario.

Taking Figure 1 as the example, the first training instance consists of text that faithfully describes the information in the data without hallucination, that is, the information in the text can all be inferred from the data. However, the other training instances contain hallucinations in their text. For instance, the second training instance contains "1940" in the text, while the input data does not have such information, and the last instance contains two hallucinations: "Dutch", "professional racing cyclist" and one incorrect expression "October". Previous neural models view each training instance in the corpus as

the gold standard; thus, they cannot distinguish the quality of training instances and can be misguided by the low-quality ones.

The state-of-the-art DSG (abbreviation for Distant Supervision Generation) framework [6] in this setting attempted to denoise the generation model by predicting what words in the vocabulary were supported by data. Then, it penalized those words with low supportiveness when generating text. However, in the process of training, it still views all texts in the training corpus, including those low-quality ones, as the gold standard.

To address the aforementioned "noise" in the distantly-supervised data-to-text generation corpus, we first introduce a small amount of heuristically better-aligned training instances for data-to-text generation as an oracle to guide the generation model to better select more faithful training instances and reduce the weights of those noisy instances. In detail, we first use a heuristic rule to give each instance a weight of faithfulness and select a subset of training instances with high weight as the oracle training data.

Then, we perform the meta gradient descent to adjust the instance weights by minimizing the MLE loss of generation on the oracle training subset. Figure 2 demonstrates the optimization procedure. To our best knowledge, we are the first to alleviate noise in the distantly-supervised data-to-text task via meta learning. In addition, we rewrite the noisiest training instances with a low weight with a fairly trained text generator, turning them into better-aligned training instances. In this way, the model can generate more faithful content with a higher-quality training corpus.

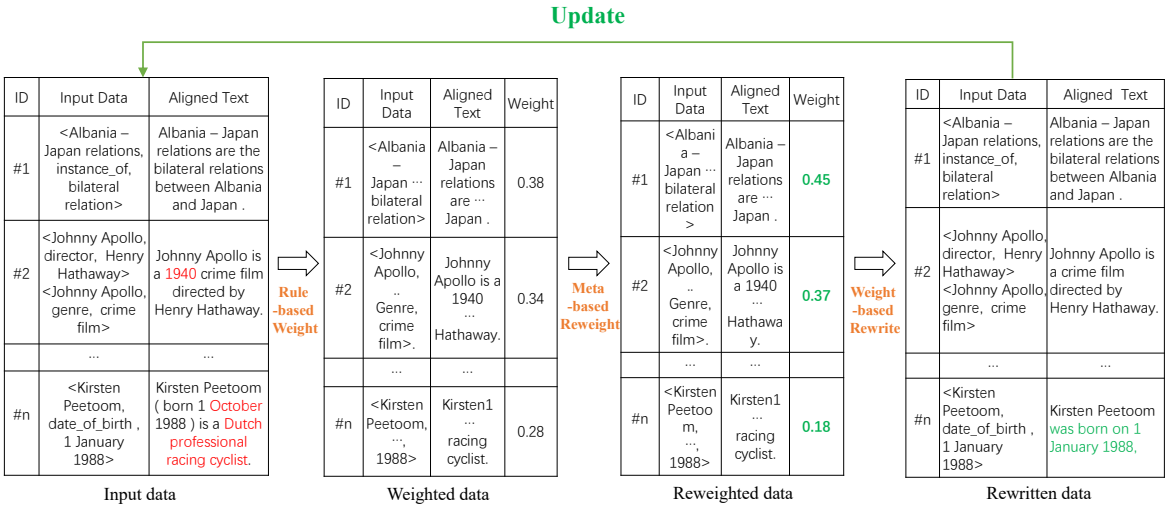

Figure 2. An illustration of the step-by-step optimizing process, given the noisy training instances. It consists of two modules: the corpus reweight module (i.e., rule-based weight and meta-based reweight) and the corpus rewrite module (i.e., weight-based rewrite). For a given batch of data, we first use heuristic rules to determine the weight of each training instance. Then, we use meta learning to further optimize those weights, given the model's performance on an oracle training subset, which consists of better-aligned training instances. In the end, we rewrite those poorly-aligned training instances, indicated by the low weight, into better ones with the trained model.

Finally, we construct a new dataset, called DIST-ToTTo, to evaluate the models' ability to generate faithful text while training on the noisy corpus. Specifically, this dataset derives from ToTTo [8], which is constructed by first crawling text and then employing human annotators to filter hallucination from the crawled text. While this human-in-the-loop construction method can produce a high-quality corpus, it requires a great deal of resources, making it difficult to expand domains.

In order to ease the burden of annotators to the full extent, we collect the crawled text as training examples and automatically extract triples from the text, following the extracting process in Fu et al. [6]. We conduct experiments on both the WITA and DIST-ToTTo datasets and apply our training procedure to both the strong base model and the

current state-of-the-art model DSG. Those experiments show our approach's effectiveness in dealing with noise in a training corpus and can help existing models to enhance their ability to generate higher fidelity text.

The contributions of this paper can be summarized as follows:

- A meta-learning-based corpus reweight module for distantly-supervised data-to-text generation is proposed to alleviate the negative impact on training a neural data-to-text model with a noisy training corpus.
- A corpus rewrite module, which reduces noise in low-quality training instances, is introduced to provide a corpus with better fidelity for training the neural data-to-text model.
- A new distantly-supervised data-to-text generation corpus, called DIST-ToTTo, is constructed. Evaluation results on both WITA and DIST-ToTTo demonstrate that our proposed corpus reweight and rewrite modules can boost both the base model and SOTA's performance in generating faithful text.

2. Related Work

2.1. Data-To-Text Generation

Data-to-text generation task has two major types of methods: pipeline-style systems [9,10] and end-to-end models [2,3]. The former decouples the generation process into sequential stages, including the content planning stage, which selects and orders the important information from the input and surface realization, which convert the content plan from the previous stage into natural language [9,10]. The latter entangles all stages and generates text directly from structured data via a neural sequence-to-sequence framework.

In recent years, neural end-to-end data-to-text generation models [2,3] have become the main framework for this task with the help of high-quality datasets. Some explore how to enhance the content selection ability of the neural sequence-to-sequence model [3,12]. The first proposes a pre-selector for selecting data. The second one uses the pointer network [25] to select important data and then uses an encoder–decoder model to generate text. Some studies have explored how to improve the fidelity of text [13,14].

The first one used optimal transport to measure the information distance between data and text, while the other attempted to exploit a pretrained language model to enhance fidelity. Another line of work [15,18,19] is to incorporate reasoning capacity into the neural data-to-text model to generate more informative and accurate text. The first two papers proposed challenging datasets that specifically require the model to perform logical reasoning for generating text.

The third one explored the use of self training to automatically generate a logical form that can teach the model how to reason. Recently, few-shot data-to-text generation [20–23] has attracted increasing attention. Some incorporate memory module [26,27], while some exploit data augmentation based on existing training set [28,29]. Kasner and Dusek [30] explored a zero-shot setting that utilizes a handcrafted template to transform triples into textual facts and uses neural modules to plan and form the final text.

Apart from the works above that require a well-aligned training dataset or handcrafted template, Fu et al. [6] proposed a partially aligned data-to-text generation task that can automatically construct the corpus, providing an efficient measure for training a data-to-text system in new domains. They sampled text from Wikipedia and automatically retrieved structured data with the knowledge base wikidata. This enables data-to-text generation for domains that lack a well-aligned dataset. The main challenge lies in dealing with the noise in the automatically constructed corpus, that is, text may contain information that is not in the input data.

We evaluated our model on the benchmark dataset proposed by Fu et al. [6], but the method can be easily extended to other domains when training on corresponding domain-specific datasets.

2.2. Meta Learning

Meta learning [31], also known as learning to learn, is one of the potential techniques for enabling an artificial agent to mimic a human's ability in using past experience to quickly adapt to unseen situations. In meta learning, the training corpus is split into support sets and query sets. In order to learn the ability of fast adaptation, the model is "trained" on the support set first, also known as the inner loop, allowing the model to adapt to the new data. Then, the model is "evaluated" on the query set, also known as the outer loop, to assess its performance on the new task. During training, by first training the model on the support set and then minimizing the model's loss function on the query set, the model is trained to adapt to new tasks through second-order gradients.

Meta learning [32] has been applied widely, especially in domain adaption [31,33,34]. As in the context of improving the model's ability of better domain adaption, the main idea is to learn the general model parameters, which consists of general features that are suitable to most tasks, and then to quickly adapt to new tasks when finetuned on a batch of data of the new task. Section 2.2 has more information regarding this.

The closest related works that involve meta learning are data reweight applications via meta learning in image classification [35,36], relation classification [37] and named entity recognition [38]. To the best of our knowledge, we are the first to alleviate the noise in distantly-supervised data-to-text tasks via meta learning. In addition, we propose to rewrite the noisiest training instances, as deemed by the model, in order to provide a higher-quality training corpus for training a more faithful neural data-to-text generation model.

In this paper, we heuristically construct an oracle training subset, consisting of better-aligned training instances, as detailed in Section 4.1. We regard this oracle training subset as the query set and regard the noisy training corpus as the support set. In the inner loop, we train the text generator on the noisy training corpus with learnable weights for each training instance, and, in the outer loop, we learn the weight for each training instance with respect to the model's performance on the oracle training subset.

Following previous work [37,39], we regard each batch of training instances as a task and push the weight of each training instance to a direction in which the outer loop with an oracle training subset has better performance (i.e., lower loss). As a result, given a batch of instances' weight to be optimized for, we associate a higher or lower weight based on the second-order gradients produced by the outer loop with an oracle training subset and the inner loop with the noisy corpus. The idea is that the increase or decrease in weights for each noisy training instance is based on whether they can maximize the performance (i.e., minimize loss) of the model on the oracle training subset.

Particularly, the updating process for our weight parameters can be explained by training instances that better comply with the meta-data knowledge (i.e., oracle training subset) will be improved, while those violating such meta-knowledge will be suppressed. This tallies with our common sense on the problem: we should reduce the influence of those highly noisy training instances while emphasizing the well-aligned ones. In the previous DSG (abbreviation for Distant Supervision Generation) framework, the disagreeing instance can adversely affect the optimization of the text generator. However, our method can alleviate its adverse effects by reducing the weight of such instances.

3. Background

3.1. Task Definition

For the data-to-text task, we can formulate each training instance as a pair of structured data and the corresponding text $E = (D, T)$. Structured data take the form of multiple triples, which can be formulated as follows: $D = \{r_i\}_{i=1}^{N}$. Each triple can be seen as $\{r_i\} =< re_i, rt_i, rv_i >$. The re_i represents the entity's name, rt_i means the type of this information, and rv_i is the corresponding value. Please note that re_i, rt_i and rv_i can all be viewed as a sequence of words. The model needs to learn to generate text $T = \{y_1, y_2, ..., y_L\}$ to faithfully and fluently describe all the information in the structured data. N represents the number of records and L represents the number of words in the text.

However, for the distantly-supervised data-to-text task discussed in this paper, the training dataset was constructed by extracting triples from text automatically using the information-extraction method. Thus, the training dataset contains "noise", since the extraction method could miss some information in the text, so the structured data in the automatically constructed training corpus may not cover all the information in the reference text. In this paper, we define "noise" as the words or phrases describing the information that are in the reference text but are absent in the automatically extracted structured data.

3.2. Base Models

In this paper, we apply our approach to two models: S2ST (abbreviation for Sequence-to-Sequence Transformer) and the state-of-the-art DSG (abbreviation for Distant Supervision Generation) model.

S2ST: during training, given structured data D and their corresponding text T, the model is expected to maximize the conditional probability:

$$P(T|D) = \prod_{t=1}^{L} P(y_t|y_{<t}, D) \tag{1}$$

t is the timestep of the decoder. We choose a Sequence-to-Sequence structure, consisting of a Transformer [40], as one of the base models. It is an attention-based model which has been proven effective in many tasks, including the distantly-supervised data-to-text task [6]. The Transformer model has two modules: transformer encoder and transformer decoder. The former represents the data with multi-layer self-attention-based structure:

$$H_k = Enc(D) \tag{2}$$

Then, the decoder uses the input feeding technique to take the previous target text as input, attending to related information in the data representation H_k and generating text word-by-word. During training, it uses the negative log-likelihood of the target text as the objective function. By minimizing the negative log-likelihood, the model can learn how to generate text with reference text as the example. More details can be found in Vaswani et al. [40].

DSG: a DSG (abbreviation for Distant Supervision Generation) framework [6] is proposed to tackle the task. Their framework can deal with the challenging over-generation problem when training on the distantly-supervised data. It first trains an estimator to calculate each word's supportiveness in the target sentence with respect to the input data, i.e., how likely the word is conveyed by the input triples. The estimator will produce a supporting matrix S where $S_{i,j}$ represents the supportiveness of the ith word in the data that support the jth word in the text.

The supportiveness score vector is aggregated from S [6] as below:

$$s_j = \log \sum_{i=1}^{|N|} \exp(S_{i,j}), \tag{3}$$

where s_j is the jth element of the vector $s \in \mathbb{R}^m$, and it stands for input data N's supportiveness to the jth word in text. Then, the framework employs a S2S (abbreviation for Sequence-to-Sequence) neural model to encode the input data and generates the descriptive text accordingly. In the training procedure, a supportiveness adaptor is used to adapt the estimated supportiveness into the loss function, while in the generation procedure, a rebalanced beam search is used to generate text with the supportiveness scores.

4. Approach

As shown in Figure 3, we use a heuristic structured data extractor to automatically construct data-to-text generation datasets, given text reports only. After training the text

generator, we can use it to automatically generate the text report for the structured data collected from the stock market, knowledge base, etc.

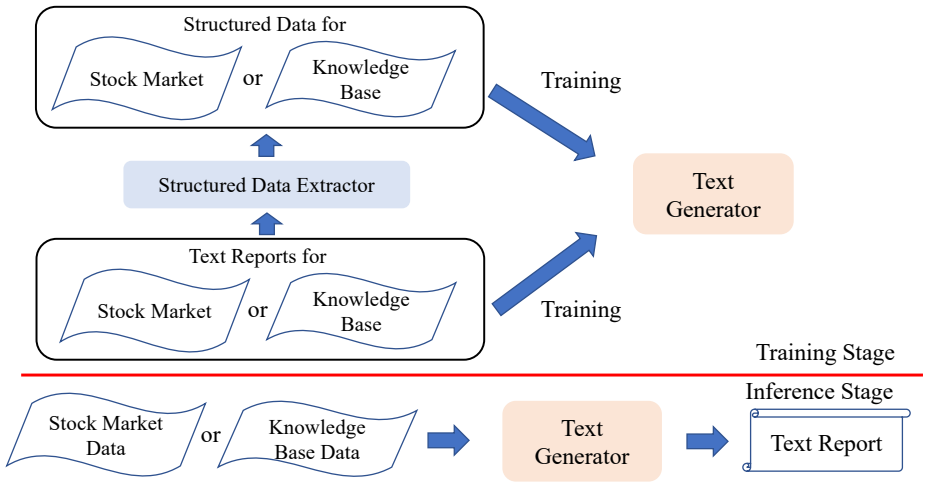

Figure 3. The functional framework of distantly-supervised data-to-text generation. In the training stage, given text reports for different sources, such as the text for the stock market or the text for the knowledge base, a heuristic structured data extractor is employed to automatically extract structured data from text. Then, the paired data instances can be used to train a text generator. In the inference stage, given structured data, the text generator can automatically generate text reports to help users better understand the meaning behind those data.

Figure 4 shows the overall training procedure. Dealing with the noise in the automatically constructed corpus, as described in Section 3.1, we propose the following three modules to address the challenge:

- In Section 4.1, we construct an oracle training subset, which can provide guidance on what kind of training instances are of high quality for the model.
- In Section 4.2, we propose a corpus reweight module, which utilizes meta learning to dynamically adjust the training instances' weight during training, in order to mitigate the negative impact of those low-quality training instances.
- In Section 4.3, we propose a corpus rewrite module that transforms the noisiest data-text training pairs into better-aligned ones, guiding the model to generate text more faithfully.

4.1. Oracle Training Subset Construction

We construct a small set of D_{oracle}, which has better alignment between data and text. It is considered a standard that can be used to evaluate other training instances' quality. First, we define a data quality confidence score cs_i. Given a data-text pair, we regard the noun phrases in the text as candidates and construct a set $candidateSet_i$. Then, we try to string match each candidate noun phrase against those in the data. If a match is found, then the information reflected by the noun phrase in the text is considered consistent with the data. If not, it is considered a hallucination. The data quality confidence score is defined as follows:

$$cs_i = \frac{|candidateSet_i \in NP(D_i)|}{|candidateSet_i|} \tag{4}$$

Then, we include training instances with data quality confidence score higher than the threshold to D_{oracle}. The cs_i can also initialize the weight of training instances for the corpus reweight module.

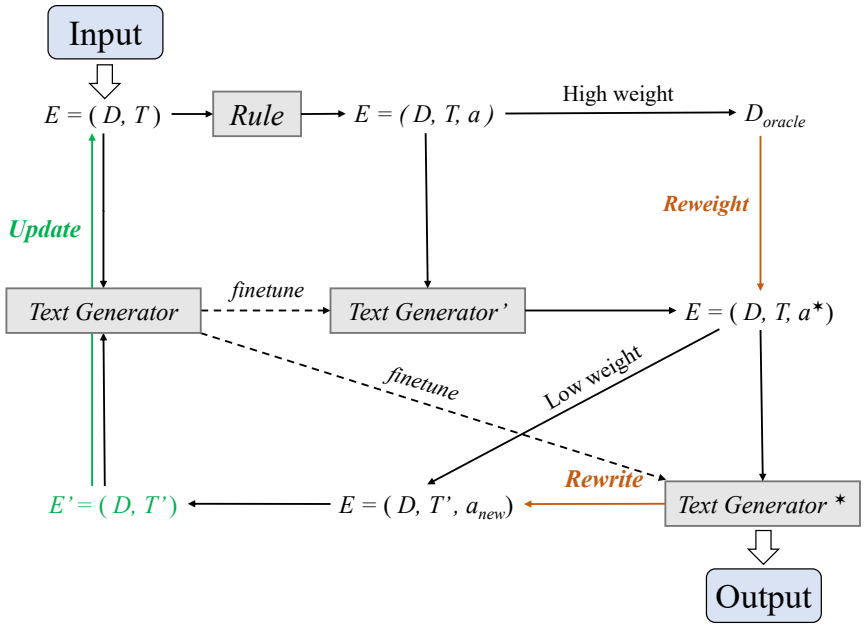

Figure 4. Implementation of our approach, which consists of corpus reweight module and corpus rewrite module. The former uses meta objective function to learn weights for training instances, while the latter rewrites those training instances with low weight.

4.2. Corpus Reweight Module

Generally, a well-aligned training instance can help the model learn how to write a text report faithfully based on structured data, while a low-quality training instance may misguide the model and let it learn to generate hallucinated expressions. We leverage the D_{oracle} using meta-learning-based corpus reweighting algorithm to increase weight for well-aligned instances while reducing weight for low-quality ones. As shown in Figure 4, the corpus reweight module can be decomposed into two major optimization steps: the first step is using meta gradient descent to reweight the corpus and the second step is to use the weights for optimizing text generation model with respect to its MLE loss. The whole training process iterates between corpus reweighting and text generator optimization.

In this paper, we utilize online meta learning [37] for data-to-text generation, which dynamically learns the corpus weights for each batch of training data via second-order derivative. In the corpus reweighting phase, given a batch of training instances D_{batch} and the weight vector a, which is initialized by the quality measure, as defined by Equation (4), we update the new set of weights as follows:

$$\theta^{'}(a^{'}) = \theta(a) - \beta\nabla_{\theta(a)}(\sum_{(D_i,T_i)\in\mathbb{D}_{\text{batch}}} a_i\mathcal{L}_{gen}(f_{\theta(a)}(D_i), T_i)) \tag{5}$$

$\mathcal{L}_{gen}$ is defined Equation (10). $f_{\theta(a)}$ represents the text generator with parameter $\theta(a)$ and $\theta(a)$ refers to the text generator's parameter with weights a for training instances in the batch. We update $\theta(a)$ to a new text generator with new parameters and weights $\theta^{'}(a^{'})$ in temporary as Equation (5). Please note that the temporary $\theta^{'}(a^{'})$ is only used to optimize corpus weights.

Based on the temporary text generation model $\theta^{'}(a^{'})$, the meta learning loss $\mathcal{L}_{meta}$ is the loss on the oracle training subset and is only used to update the weight a:

$$\mathcal{L}_{meta}(\theta^{'}(a^{'})) = \sum_{(D_i,T_i)\in\mathbb{D}_{\text{oracle}}} \mathcal{L}_{gen}(f_{\theta^{'}(a^{'})}(D_i), T_i) \tag{6}$$

After optimizing over the $\mathcal{L}_{meta}$, we can optimize a through second-order derivatives. Thus, the new weight vector a^* can be obtained by minimizing the meta learning loss $\mathcal{L}_{meta}$.

With the optimized instance weights a^*, we can update the text generation model according to Equation (7), which is the actual step to update the parameters of the text generator:

$$\theta^* = \theta - \beta \nabla_\theta \left(\sum_{(D_i, T_i) \in \mathbb{D}_{\text{batch}}} a_i^* \mathcal{L}_{gen}(f_\theta(D_i), T_i) \right). \tag{7}$$

In this corpus reweight module, the a is optimized so that the text generator can perform better with respect to the D_{oracle}. With such an optimization step, we could maximize the role of those training instances, which potentially have better-aligned data-text pairs.

4.3. Corpus Rewrite Module

Since low-quality training instances impose negative impacts on the model's training process, we propose to rewrite the poorly-aligned training instances with the fairly-trained text generator. We sample the training instances that its corresponding weight a_i is below the threshold 0.2, which is selected based on model's performance on the development set, for rewriting.

Since we want to provide higher-quality training instances for those sampled poorly-aligned training instances, we first pretrain a text generator with the help of the corpus reweight module, described in Section 4.2. The idea behind this is that a fairly-trained text generator is needed in order to produce better-aligned text for structured data. Then, we feed the structured data to the text generator. During generation, given the structured data, the text generator will generate the text from the vocabulary word-by-word:

$$y_i^* = \arg\max_{y_i^{'}} \prod_{t=1}^{T} P(y_{i,t}^{'} | y_{i,<t}^{'}, D_i) \tag{8}$$

Next, we determine whether the written text is better than the original text. We calculate the weight for the new text y_i^*, which can be initialized using Equation (4). By comparing the new a_i^* with the original a_i for original text y_i, we can decide which is better. If the rewritten text gets higher weight, with respect to the structured data input, we replace the original text in the corpus with this rewritten higher-quality text:

$$y_i^{new} = \begin{cases} y_i^*, & \text{if } a_g^* > a_g \\ y_i, & \text{if } a_g^* <= a_g \end{cases} \tag{9}$$

4.4. Training

Given a batch of input data $\{D\}_G$ and the corresponding text $\{Y\}_G$, we define $\mathcal{L}_{gen}$ of text generator as optimizing the MLE (Equation (10)). Please note that M_g represents the length of the text, while G represents the number of batches. Z is the normalization factor.

$$\mathcal{L}_{gen} = -\frac{1}{Z} \sum_{g=1}^{G} \sum_{t=1}^{M_g} \log P(y_{g,t}^{new} | y_{g,<t}^{new}, D_g) \tag{10}$$

4.5. Algorithm

We illustrate the iterative algorithm for model training in Algorithm 1. First, we pretrain the text generator with the original MLE loss. Then, we employ the corpus reweight module. At epoch t, We apply the online reweighting algorithm, which updates a^* and θ using batches of training data and the meta objective function (lines 3–6). Then, we rewrite the noisiest data-text pairs with the fairly trained text generator (lines 7–11).

Algorithm 1 Our approach for data-to-text generation model from noisy data.

Require: $\mathbb{D}_{\text{train}}$ (the whole training dataset), $\mathbb{D}_{\text{batch}}$ (a batch of training data), $\mathbb{D}_{\text{oracle}}$ (an oracle training subset with better-aligned training instances), f_θ (parameters of the text generation model), a^* (the weight of each training instances with meta learning ($\mathcal{L}_{meta}$, Equation (6))), cs_i (the data quality confidence score of training instances (Equation (4)))

Ensure: θ_N (the learned parameters of the text generation model after N epochs)

```
Pre-train f_θ on D_train for multiple epochs
for epoch t = 1 to N do
    for each batch D_batch in D_train do
        Minimize L_meta with Equation (6) via D_batch and D_oracle and obtain the a*
        Update text generator with a* using Equation (7)
    end for
    for each batch D_batch in D_train do
        if cs_i < threshold then
            uses the text generator to produce new reference
        end if
    end for
    Reload the training dataset with some poorly-aligned data-text pair replaced by the
    newly generated text
end for
```

5. Experiments

5.1. Setup

5.1.1. Dataset

The WITA (https://github.com/fuzihaofzh/distant_supervision_nlg, accessed on 16 September 2022) [6] dataset is split into a training set, a development set and a testing set of 50,000, 5000 and 400 instances, respectively. The testing set is manually constructed, while the training and development sets are automatically constructed. In addition, we collect the DIST-ToTTo dataset, described in Section 1, providing another testbed to assess models' performance on dealing with noisy corpus. For constructing the training set and development set, we collect the crawled version of the text from ToTTo [8] dataset and replace the noisiest 20% of raw text, measured by data quality confidence score (Equation (4)), which is inspired by Fu et al. [6], with the manually annotated version from ToTTo.

For the testing set, we collect all the text from the manually annotated version from ToTTo to assess the model's ability to generate faithful text. Since the test set of ToTTo is not publicly available, we split the original development set into the development and test set for DIST-ToTTo. The resulting dataset consists of 120,861, 3836 and 3836 examples for training, development and test set, respectively.

Table 1 demonstrates the data statistics of both the WITA and DIST-ToTTo datasets. DIST-ToTTo has 2.3 times larger corpus and the length of reference is longer than WITA. The variance of the number of input structured data (KB) is larger than WITA, while the mean and the median number of input data is comparable with WITA. Furthermore, the vocabulary size is 1.5 times larger than WITA, all suggesting that DIST-ToTTo provides a more challenging testbed for evaluating models' ability to generate faithful text when dealing with noisy data.

Table 1. Statistics of WITA (abbreviation for the data source Wikipedia and Wikidata) and DIST-ToTTo (abbreviation for Distantly-supervised Table-To-Text) datasets. KB number refers to the number of knowledge base triples, that served as the model's input, for each data instance. For the text length and KB number, the data are mean, median, min and max, respectively. The statistics of WITA are from Fu et al. [6].

	WITA	**DIST-ToTTo**
Size	55,400	128,533
Text Length	(18.8, 17, 5, 59)	(24.0, 22, 3, 81)
KB Number	(3.0, 3, 1, 11)	(3.5, 3, 1, 130)
Vocabulary	102,404	153,963

5.1.2. Evaluation Metrics

Following Fu et al. [6], we use the following evaluation metrics to assess models' performance with the evaluation script by Novikova et al. [41]:

- BLEU [42]: It evaluates the model's generated text's quality based on the n-gram overlap between generated text and reference text. The output of BLEU ranges from 0% to 100%. If the generated text is identical to the reference text, the score will be 100%. If the generated text shares no n-gram overlap with the reference text, the score will be 0%. The calculation of BLEU is illustrated in Equation (11). It consists of the following three parts. P_n is the n-gram precision of the generated text, compared to the reference. w_n is a positive weight for each n-gram. BP is the abbreviation for brevity penalty, which will penalize those generated texts that are shorter than reference, since the shorter the text, the higher precision it can potentially obtain. In practice, we report the BLEU score with N = 4 and the uniform weights $w_n = \frac{1}{4}$.

$$BLEU = \exp(\sum_{n=1}^{N} w_n \log P_n) \cdot BP \tag{11}$$

- NIST [43]: It proposes information-weighted N-grams counts that weigh more heavily to the N-grams that are deemed more informative (i.e., occur less frequently than others). The details can be found in Doddington [43].
- METEOR [44]: In addition to exact string matching between words in generated text and reference, it proposed to use WordNet to match words that share the same stem or are synonyms of each other, since those words share the same meaning. Furthermore, it proposes to group words in the text into chunks and use this to measure how well-ordered the words in the generated text are with respect to the reference. Equation (12) shows how the METEOR score is calculated. The $Fmean$ combines the precision and the recall of generated text. The $Penalty$ is based on the number of chunks in the matched sequence in the generated text. The fewer the chunks are grouped, the better the words are ordered, compared to the reference.

$$METEOR = Fmean \cdot (1 - Penalty) \tag{12}$$

- ROUGE$_L$ [45]: It proposes to use the Longest Common Subsequence (LCS) to match the generated text and reference. Equation (13) shows that $ROUGE_L$ is the F-measure of the precision and recall of the length of matched LCS. Since the β will be set to a large number, this measure is actually recall-oriented.

$$ROUGE_L = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}} \tag{13}$$

- CIDEr [46]: It proposed to use Term Frequency Inverse Document Frequency (TF-IDF) as weights to characterize the similarity between the generated text and reference and use cosine similarity function to calculate the CIDEr score. The idea of using TF-IDF is that it will give higher weight to infrequently occurring words in the dataset and

lower weight to those commonly occurring words, as the latter will be deemed less informative. The details can be found in Vedantam et al. [46].

5.1.3. Implementation Details

We follow Fu et al. [6]'s training configurations for the base model and DSG part. We chose the introduced hyper-parameters based on performance on the development set. The model is built on Fairseq [47]. We manually tune hyper-parameters on the development set and pretrain the model with MLE loss for 19 epochs from {15, 16, 17, 18, 19, 20, 21, 22, 23} on S2ST and 34 epochs from {30, 31, 32, 33, 34, 35, 36, 37, 38} on DSG for WITA dataset, before finetuing with corpus reweight module. For DIST-ToTTo dataset, the S2ST is pretrained for 26 epochs from {22, 23, 24, 25, 26, 27, 28, 29, 30 } and DSG is pretrained for 39 epochs from {35, 36, 37, 38, 39, 40, 41, 42, 43 }. The threshold for the rewrite module is set for 0.2 from {0.1, 0.2, 0.3, 0.4, 0.5}. The learning rate β is set for 0.02 from {0.01, 0.02, 0.03}. D + Full takes 12 h to train on a single Tesla V100-SXM2-16GB on the DIST-ToTTo dataset.

5.2. Results

5.2.1. Comparing Methods

We apply our approach to a base model and a state-of-the-art model with ablations. The methods for comparison are as follows:

- S2SL follows the Pointer-Generator [48] framework that employs copy mechanism and coverage to a LSTM-based [49] encoder–decoder framework. This has been a competitive model in the WebNLG [50] task.
- S2SG is a variant to S2SL, which uses GRU-based [51] encoder–decoder framework instand of the LSTM-based one in S2SL.
- S2ST employs a Transformer [40] model for the S2S (abbreviation for Sequence-to-Sequence) data-to-text generation, which is proven to be more effective than a traditional S2S model, equipped with an attention [52] and copy [53] mechanism, as in Fu et al. [6].
- DSG is the current state-of-the-art model [6] for this task. It trains an estimator to penalize unrelated words in the vocabulary based on the structured data input and uses it to rebalance the beam search.
- S2ST + Meta can be considered as an ablation study that only employs the corpus reweight module to the S2ST model.
- S2ST + Full fully employs our approach, consisting of both a corpus reweight and corpus rewrite module, to the S2ST model.
- D + Meta can be considered as an ablation study that only employs the corpus reweight module to the DSG model.
- D + Full can be considered as the full model, which employs both the corpus reweight and corpus rewrite module to the state-of-the-art model DSG.

5.2.2. Automatic Evaluation

Table 2 shows the evaluation results for applying our model to S2ST and state-of-the-art DSG on both the WITA and DIST-ToTTo datasets. In general, our approach is effective for both the base model S2ST and the state-of-the-art DSG with improvements consistently across all evaluation metrics on both the WITA and DIST-ToTTo datasets (S2ST + Full v.s. S2ST and D + Full V.S. DSG).

For example, regarding the widely used metric BLEU, our model (D + Full) improved the state-of-the-art DSG from 56.69 to 58.80 on the WITA dataset, obtaining an improvement of 3.72%. Furthermore, on the same dataset, in terms of another widely used metric CIDEr, our model (D + Full) alleviated the metric from 5.380 to 5.573, obtaining a 3.59% relative improvement. On the DIST-ToTTo dataset, our model (D + Full) achieved similar performance compared to the state-of-the-art model DSG.

We additionally have the following observations: (1) Our model's further improvement on the DSG model, which is designed to penalize unrelated words in the vocabulary according to structured data input, showed that penalizing poorly-align data-text pairs during the training process was complementary to the DSG. (2) Our approach can bring more improvements for the S2ST model compared with DSG. This is because noisy data have a worse impact on S2ST due to its lack of design to combat the noises in the distantly supervised corpus. (3) Ablations showed that both the corpus reweight module and corpus rewrite corpus contributed to the overall improvement for both models across evaluation metrics, as additionally applying the corpus rewrite module (S2ST + Full and D + Full) to a model only equipped with the reweight module (S2ST + Meta and D + Meta) further improved the performance in all metrics.

Table 2. Automatic evaluation results on both datasets' test set. We applied our approach to the base model S2ST (abbreviation for Sequence-to-Sequence Transformer), which is represented as S2ST + Full and the state-of-the-art DSG (abbreviation for Distant Supervision Generation) model, noted as D + Full. Furthermore, we include the ablation results in this paper by only applying the corpus reweight module to the corresponding model (S2ST + Meta and D + Meta). The results show that both the proposed corpus reweight and corpus rewrite modules can improve the base model S2ST and the state-of-the-art model DSG across all five metrics on both WITA (abbreviation for the data source Wikipedia and Wikidata) and DIST-ToTTo (abbreviation for Distantly-supervised Table-To-Text) datasets. Particularly, our model (D + Full) can improve the state-of-the-art model DSG by 3.72% on the WITA dataset and 3.82% on the DIST-ToTTo dataset in terms of the widely used metric BLEU (abbreviation for BiLingual Evaluation Understudy).

Dataset	Model	BLEU	NIST	METEOR	ROUGE$_L$	CIDEr
WITA	S2SL	48.08	8.459	38.75	70.92	4.415
	S2SG	47.78	8.267	38.31	72.21	4.543
	S2ST	54.52	8.631	41.85	73.81	5.045
	DSG	56.69	9.241	43.09	76.26	5.380
	S2ST + Meta	55.33	8.925	42.25	74.86	5.150
	S2ST + Full	56.34	9.097	42.75	75.71	5.255
	D + Meta	58.00	9.304	43.85	76.72	5.486
	D + Full	58.80	9.341	44.10	77.14	5.573
DIST-ToTTo	S2SL	25.25	6.555	26.23	48.32	2.140
	S2SG	24.41	6.727	25.73	47.72	2.050
	S2ST	35.91	8.343	31.64	54.89	2.697
	DSG	38.22	8.323	33.09	57.60	2.950
	S2ST + Meta	38.52	8.869	33.11	57.14	2.898
	S2ST + Full	39.10	8.931	33.82	57.61	2.944
	D + Meta	39.03	8.690	33.70	58.16	3.019
	D + Full	39.68	8.785	34.02	58.68	3.084

5.2.3. Human Evaluation

We compare the S2ST, S2ST + Full, DSG and D + Full model performance on the test set for both the WITA and DIST-ToTTo datasets in human evaluation. We sampled 50 generated texts on the test set for each model, arranged the results of those models into pairs and asked three human raters to determine which one in the pair performed better in terms of grammaticality (which one is more fluent and grammatical?) and fidelity (which one is more faithful to the data?).

If the human raters deem those texts generated from two different models are of the same quality, they are allowed to assign a 0.5 score to each of the two models. Each example was evaluated by three different graduates, who are proficient in English, and we report the average of three human raters' results in Table 3. The reported result is the subtraction of the percentage of time a system is considered better and when considered worse. It can be

seen that our model can help improve the S2ST and DSG in terms of both grammaticality and fidelity on both the WITA and DIST-ToTTo datasets, respectively.

For example, on the WITA dataset, our model (D + Full) obtained scores 12.67 and 11.33, in terms of grammaticality and fidelity, respectively. Both exceed the state-of-the-art model DSG's 11.33 and 7.44, respectively. On the DIST-ToTTo dataset, these conclusions still stand in terms of both grammaticality and fidelity. These results support our claim that reweighting training corpus based on each data-text pair's quality via meta learning and rewriting those poorly-aligned data-text pairs during training can help the model learn to generate more faithful text. Overall, our model can produce faithful text while keeping relatively good grammaticality.

Table 3. Human evaluation results on WITA (abbreviation for the data source Wikipedia and Wikidata) and DIST-ToTTo (abbreviation for Distantly-supervised Table-To-Text) datasets. The score in the table is calculated as the percentage of time the model is considered better minus the percentage of time it is considered worse. We use Kappa [54] to evaluate the agreement between evaluators. On the WITA dataset, the Kappa values on grammaticality and fidelity are 0.4 and 0.56, respectively. On the DIST-ToTTo dataset, the Kappa values on grammaticality and fidelity are 0.45 and 0.52, respectively.

Dataset	Model	Grammaticality	Fidelity
WITA	S2ST	11.61	5.72
	S2ST + Full	16.17	12.06
	DSG	11.33	7.44
	D + Full	12.67	11.33
DIST-ToTTo	S2ST	7.72	−2.67
	S2ST + Full	11.67	7.5
	DSG	15.06	11.28
	D + Full	20.00	22.78

5.3. Analysis

5.3.1. Over-Generation Error Analysis

Following Fu et al. [6], we quantitatively analyzed the over-generation problem by counting the over-generated n-gram tokens. In detail, given generated sentences, we first filtered the stopwords. Then, we counted those remaining n-gram tokens if they do not appear in the structured data and report the statistics in Table 4. Lower over-generation tokens indicate that text is more faithful to data. Our approach can improve both the S2ST and DSG models across all four n-grams statistics on both datasets. On WITA, our model can reduce 8.8% over-generated 1-gram for the S2ST model (down from 624 to 569) and 6.2% over-generated 1-gram (down from 463 to 434) for the DSG model.

On DIST-ToTTo, the reduction percentages are 18.1% and 9.1%, respectively. Our model (D + Full or D + Meta) exceeds the state-of-the-art model DSG across all kinds of n-gram metrics. We hypothesize that this is because reducing the low-quality data-text pairs' negative impact during training can alleviate the model's burden that being forced to learn to generate hallucinations. This automatic evaluation result aligns with the human evaluation's fidelity test. Both show that our model can enhance the model's ability to generate faithful text.

5.3.2. Noise Effect Analysis

Since the dataset contains noise among its training instances, the straightforward way to avoid the noise is to drop the noisiest data from the dataset, avoiding the model from being misguided by them. Using the data quality confidence score for each training instance, we can filter those with the lowest score. Take the last instance in Figure 2 as the example, the input information is <Kirsten Peetoom, data_of_birth, 1 January 1988> and

the corresponding text is Kirsten Peetoom (born 1 October 1988) is a Dutch professional racing cyclist.

Table 4. N-gram statistics for over-generation error analysis. The score reflects the number of ngram tokens in the text (excluding stopwords) that do not appear in the structured data, which can potentially be seen as hallucination. While this analysis provides an indication of models' performance on generating faithful text, which is supplementary to the fidelity metric in human evaluation (Table 3), it cannot guarantee that reported ngram tokens all contribute to hallucination. Lower over-generation tokens indicate that text is more faithful to data.

Dataset	Model	1-Gram	2-Gram	3-Gram	4-Gram	5-Gram
WITA	S2ST	624	2232	2804	2770	2492
	S2ST + Meta	588	2158	2733	2693	2406
	S2ST + Full	569	2110	2671	2625	2343
	DSG	463	2041	2635	2594	2309
	D + Meta	431	1998	2606	2590	2317
	D + Full	434	1970	2584	2572	2294
DIST-ToTTo	S2ST	14234	27539	28954	26320	22769
	S2ST + Meta	12096	25496	27249	24832	21343
	S2ST + Full	11646	24927	26880	24538	21081
	DSG	10183	24437	26659	24473	21098
	D + Meta	9336	22620	24629	22362	18954
	D + Full	9253	22426	24479	22259	18863

Since the text contains incorrect information and hallucination, which means it contains information that is not in the input, it has a low data quality confidence score. We rank the training instances based on the data quality confidence score and purge different percentages of those poorly-aligned training instances with the lowest data quality score. We compare the results with the S2ST model trained on the full training set (0%) on the WITA dataset in Table 5. It shows that by purging 10% of them, the model is slightly improved on NIST (+2.62% from 8.631 to 8.857), ROUGE$_L$ (+0.18%, from 73.81 to 73.94) and CIDEr (+1.76%, from 5.045 to 5.134) without interference from those noises.

However, after dropping more poorly-aligned training instances, the model's performance drops instead. With only 50% remaining data for training, the BLEU dropped significantly (-14.47%, from 54.52 to 46.63). We hypothesize that while dropping those noisy data can alleviate misguidance, they also hinder the model's ability to learn how to phrase the text with the limited dataset. Our model (S2ST + Full in Table 2), which is applied to S2ST, performed better than any settings of dropping noisy data for S2ST on all five metrics. This indicates our model's ability to alleviate the negative impact of the noisy data.

Table 5. Noise Effect Analysis on the WITA (abbreviation for the data source Wikipedia and Wikidata) dataset. We drop those training instances with the least quality, measured by the data quality confidence score and train the S2ST (abbreviation for sequence-to-sequence transformer) model on the dropped datasets, respectively.

Drop Percentage	BLEU	NIST	METEOR	ROUGE$_L$	CIDEr
0%	54.52	8.631	41.85	73.81	5.045
10%	53.20	8.857	41.27	73.94	5.134
20%	52.79	8.734	41.37	73.70	5.006
30%	51.69	8.675	40.69	72.97	4.852
40%	48.81	8.479	39.15	71.12	4.602
50%	46.63	8.209	37.98	69.75	4.412
S2ST + Full	56.34	9.097	42.75	75.71	5.255

5.3.3. Case Study

We compare our approach's results against two strong models on both the WITA and DIST-ToTTo datasets in Tables 6 and 7. In general, our model can faithfully illustrate information given in data with less hallucination that cannot be inferred from data.

Our approach has nice properties in these cases: (1) It does not include hallucination that cannot be inferred from data, while S2ST or DSG tends to generate some (e.g., "2003", "2012" in the first example in Table 6), marked in red; (2) Our model can sometimes better describe information from the structured data with less missing information, while S2ST misses the type of painting in the second example in Table 6; (3) Our model, especially when applied to the state-of-the-art DSG, generates text with high fidelity. For instance, DSG generates incorrect expressions in the third example in Table 6, while our model does not. Results on the DIST-ToTTo dataset (Table 7) share similar conclusions above.

Table 6. Case study on the WITA dataset. The **bold** font stands for hallucination that cannot be inferred from data. The [MI] ([missing information]) indicates that some information in the data is not illustrated in the text and the *italic* font stands for incorrect or repetitive expressions.

KB	S2ST	S2ST + Full	DSG	D + Full	Gold
⟨The Keys of the Kingdom, author, A. J. Cronin⟩, ⟨The Keys of the Kingdom, genre, novel⟩	The Keys of the Kingdom is a **2003** novel by **American** author A. J. Cronin.	The Keys of the Kingdom is a novel by A. J. Cronin.	The Keys of the Kingdom is a **2012** novel by A. J. Cronin.	The Keys of the Kingdom is a novel by A. J. Cronin.	The Keys of the Kingdom is a novel by A. J. Cronin.
⟨The Roaring Forties, creator, Frederick Judd Waugh⟩, ⟨The Roaring Forties, material_used, oil paint⟩, ⟨The Roaring Forties, inception, 1908⟩	The Roaring Forties is a [MI] painting by Frederick Judd Waugh in 1908.	The Roaring Forties is a 1908 oil painting by Frederick Judd Waugh.	The Roaring Forties is a 1908 [MI] painting by Frederick Judd Waugh.	The Roaring Forties is a 1908 oil painting by Frederick Judd Waugh.	The Roaring Forties is a 1908 oil painting by Frederick Judd Waugh.
⟨Murdoch Cameron, date_of_birth, 31 March 1847⟩, ⟨Murdoch Cameron, date_of_death, 28 April 1930⟩, ⟨Regius Professor of Obstetrics and Gynaecology, Glasgow, part_of, University of Glasgow⟩, ⟨Murdoch Cameron, employer, University of Glasgow⟩	Murdoch Cameron (31 March 1847–28 April 1930) was an **English** professor of *Glasgow* at the University of Glasgow.	Murdoch Cameron (31 March 1847–28 April 1930) was [MI] Professor of Obstetrics and Gynaecology at the University of Glasgow.	Murdoch Cameron (31 March 1847–28 April 1930) was a [MI] Professor of *Glasgow* at the University of Glasgow.	Murdoch Cameron (31 March 1847–28 April 1930) was Regius Professor of Obstetrics and Gynaecology at the University of Glasgow.	Murdoch Cameron (31 March 1847–28 April 1930) was Regius Professor of Obstetrics and Gynaecology at the University of Glasgow.

Table 7. Case study on DIST-ToTTo dataset. The **bold** font stands for hallucination that cannot be inferred from data. The [MI] ([missing information]) indicates that some information in the data is not illustrated in text and the *italic* font stands for incorrect or repetitive expressions.

KB	S2ST	S2ST + Full	DSG	D + Full	Gold
⟨Title: Marc Abaya, Television⟩, ⟨Television: Ligaw na Bulaklak⟩, ⟨Television: Francis⟩, ⟨Television: ABS-CBN⟩	**In 2011**, Marc Abaya appeared [MI] in ABS–CBN's Ligaw Bulaklak.	Marc Abaya played the role of Francis in ABS–CBN's Ligaw na Bulaklak.	Abaya played Francis in [MI] Ligaw na Bulaklak.	Marc Abaya played as Francis in ABS–CBN 's Ligaw na Bulaklak.	Abaya played the role of Francis in an ABS–CBN, Ligaw na Bulaklak.
⟨Title: Serbia in the Junior Eurovision Song Contest, Participation⟩, ⟨Year: 2007⟩, ⟨Artist: Nevena Božović⟩, ⟨Song: "Piši mi"⟩	In 2007 [MI], Serbia selected Nevena Božović with the song "Pišši mi *mi*".	Nevena Božović represented Serbia at the 2007 [MI] contest with the song "Piši mi".	Nevena Božović represented Serbia in the Junior Eurovision Song Contest 2007 [MI].	Nevena Božović represented Serbia in the Junior Eurovision Song Contest 2007 with the song "Piši mi".	At the 2007 Junior Eurovision Song Contest, Nevena Božović represented Serbia with the song "Piši mi".
⟨Title: The Weight of These Wings, Awards⟩, ⟨Year: 2017⟩, ⟨Association: ACM Awards⟩, ⟨Category: Album of the Year⟩, ⟨Result: Won⟩	The Weight of These Wings was **nominated** for Album of the Year at the 2017 ACM *M* Awards.	The Weight of These Wings won Album of the Year at the 2017 ACM Awards.	At the 2017 ACM Awards, the album [MI] won Album of the Year.	At the ACM Awards of 2017, The Weight of These Wings won Album of the Year.	The Weight of These Wings won Album of the Year at the 2017 ACM Awards.

5.3.4. Rewrite Analysis

We demonstrate cases of rewriting the data-text training instance, which is deemed low quality in the training process (Tables 8 and 9) on both the WITA and DIST-ToTTo datasets. Due to the limitation of the information extractor [6] that is used to automatically construct the corpus, the original text of the training instance contains hallucinate expression (e.g., "(7 February 1948–11 March 2015)") or incorrect expression (e.g., "keyboard player and composer"), as shown in the first example in Table 8, compared to the structured data. Illustrated by these examples, our corpus rewriting module can help cleanse these kinds of expressions and produce a better-aligned data-text training instance. For example, it removes the hallucinate expressions in bold from the original text for all three examples and further corrects the expression in italics of the first example.

Table 8. Rewrite case for D + Full on the WITA Dataset. The **bold** font stands for hallucinate expressions that cannot be inferred from data, and the *italics* font stands for incorrect expressions that contradict the data.

KB	Original Text	Rewritten Text
⟨James Boyd, occupation, American football player⟩, ⟨James P. Boyd, position_held, member of the Ontario Provincial Parliament⟩	James Boyd *Greenspoon* **(7 February 1948–11 March 2015)** was an American *keyboard player and composer*, best known as a member of the *band Three Dog Night*.	James Boyd is an American former football player and member of the Ontario Provincial Parliament.

Table 8. *Cont*.

KB	Original Text	Rewritten Text
⟨Amanikhabale, position_held, King of Kush⟩, ⟨King of Kush, organization_directed_from_the_office, Kush⟩	Amanikhabale (also transliterated Astabarqaman) was a King of Kush **(circa 50 BCE- 40 BCE)**.	Amanikhabale was the King of Kush for Kush.
⟨MARCbot, manufacturer, Exponent⟩, ⟨MARCbot, subclass_of, military robot⟩	The Multi—function Agile Remote - Controlled Robot (MARCbot) is a military robot created by Exponent Inc. for the **United States Army Rapid Equipping Force**.	The MARCbot was a military robot of the Exponent.

Table 9. Rewrite case for D + Full on DIST-ToTTo Dataset. The **bold** font stands for hallucinate expressions that cannot be inferred from data, and the *italic* font stands for incorrect expressions that contradict the data. The [MI] ([missing information]) indicates that some information in the data is not illustrated in the text.

KB	Original Text	Rewritten Text
⟨Title: List of heads of government of Sierra Leone, Prime Ministers of Sierra Leone⟩, ⟨Prime Minister, Name (Born–Died): Sorie Ibrahim Koroma (1930–1994)⟩, ⟨Term of Office, Took Office: 21 April 1971⟩, ⟨Term of Office, Left Office: 8 July 1975⟩	From **1968** until **1985**, Koroma served various functions in the government, including **Minister of Agriculture and Natural Resources (1969–1971)**, **Vice–President (1971–1985)** and Prime Minister (1971–1975), **Minister of Finance (1975–1978)**.	Koroma was the Prime Minister of Sierra Leone from 21 April 1971 to 8 July 1975.
⟨Title: List of mayors of Manchester, 1838–1893⟩, ⟨Mayor: Sir James Watts⟩, ⟨Tenure began: 1855⟩, ⟨Tenure ended: 1857⟩	Sir James Watts JP **(6 March 1804–7 April 1878)** was Mayor of Manchester (1855–1857), **High Sheriff of Lancashire and owner of Abney Hall**.	Sir James Watts was Mayor of Manchester from 1855 to 1857.
⟨Title: List of the oldest mosques, Eurasia⟩, ⟨Building: Masjid Mazin⟩, ⟨Country: Oman⟩	Masjid Māzin is considered to be the oldest mosque in the country [MI].	Masjid Mazin is the oldest mosques in Oman.

6. Conclusions

Despite the success of data-driven deep-learning models for data-to-text generation, they require large-scale, high-quality data for training, limiting their ability to generalize to new domains without sufficient training data. The distantly-supervised data-to-text generation's ability to automatically generate training instances makes generalization easier. However, it inevitably induces an over-generation problem: text includes hallucination.

To guide the model to generate more faithful text, we proposed three modules: (1) we heuristically constructed an oracle training subset consisting of better-aligned training instances. This can provide a standard for the model to know what constitutes a good training instance. (2) We dynamically increased the weights of those well-aligned instances and reduce the weights of low-quality ones with meta learning. (3) We rewrote those low-quality training instances with the fairly-trained text generation model, providing the model with better supervision signals to learn how to faithfully generate text.

The automatic evaluation, human evaluation and multiple analyses on both WITA (abbreviation for the data source Wikipedia and Wikidata) and DIST-ToTTo (abbreviation

for Distantly-supervised Table-To-Text) datasets showed that our model could improve both the basic S2ST (abbreviation for Sequence-to-Sequence Transformer) model and the state-of-the-art DSG (abbreviation for Distant Supervision Generation) model in terms of fluency and fidelity. In particular, our model improved upon the state-of-the-art DSG model by 3.72% on the WITA dataset and by 3.82% on the DIST-ToTTo dataset in terms of the widely used metric BLEU (abbreviation for BiLingual Evaluation Understudy).

In the future, there are some possible directions to further improve the performance of the distantly-supervised data-to-text generation model. (1) Explore a better strategy to create an oracle training subset so that meta learning can provide more precise guiding signals for training. (2) Explore how to automatically produce better training instances with dual learning. (3) Explore how to integrate domain knowledge to generate better text.

Author Contributions: Conceptualization, H.G. and X.F.; methodology, H.G.; software, H.G.; validation, H.G., X.F. and B.Q.; formal analysis, H.G.; investigation, H.G.; resources, H.G.; data curation, H.G.; writing—original draft preparation, H.G.; writing—review and editing, H.G., X.F. and B.Q.; visualization, H.G.; supervision, X.F. and B.Q.; project administration, X.F. and B.Q.; funding acquisition, B.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Key R&D Program of China via grant 2020AAA0106502, National Natural Science Foundation of China (NSFC) via grant 62276078 and the Province Key R&D Program of Heilongjiang via grant 2022ZX01A32.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data included in this study are available upon request by contact with the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following variables are used in this manuscript:

E	A training instance consists of the model's input and target.
D	Structured data as model's input.
T	Text about the structured data, which is the target of the model
r_i	A triple, representing one of the structured data.
re_i	Entity's name.
rt_i	The type of this information.
rv_i	The value of the triple.
y_t	One word in the target text.
S	The supporting matrix in a DSG (Distant Supervision Generation) model.
s	DSG's aggregated supportiveness score vector.
$candidateSet_i$	Noun phrases in the target text.
cs_i	Data quality confidence score.
D_{oracle}	An oracle training subset with better-aligned training instances.
D_{batch}	A batch of training data.
f_θ	Parameters of the text generation model.
a^*	The weight of each training instance with meta learning.
β	Learning rate.

References

1. Lebret, R.; Grangier, D.; Auli, M. 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, Austin, TX, USA, 1–5 November 2016; pp. 1203–1213.
2. Wiseman, S.; Shieber, S.; Rush, A. Challenges in Data-to-Document Generation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 2253–2263.
3. Puduppully, R.; Dong, L.; Lapata, M. Data-to-text Generation with Entity Modeling. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July 2019; pp. 2023–2035.

4. Uehara, Y.; Ishigaki, T.; Aoki, K.; Noji, H.; Goshima, K.; Kobayashi, I.; Takamura, H.; Miyao, Y. Learning with Contrastive Examples for Data-to-Text Generation. In Proceedings of the 28th International Conference on Computational Linguistics, Online, 8–13 December 2020; pp. 2352–2362.
5. Guo, B.; Wang, H.; Ding, Y.; Wu, W.; Hao, S.; Sun, Y.; Yu, Z. Conditional text generation for harmonious human–machine interaction. *ACM Trans. Intell. Syst. Technol. (TIST)* **2021**, *12*, 1–50. [CrossRef]
6. Fu, Z.; Shi, B.; Lam, W.; Bing, L.; Liu, Z. Partially-Aligned Data-to-Text Generation with Distant Supervision. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 9183–9193.
7. Chen, D.L.; Mooney, R.J. Learning to Sportscast: A Test of Grounded Language Acquisition. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 128–135.
8. Parikh, A.; Wang, X.; Gehrmann, S.; Faruqui, M.; Dhingra, B.; Yang, D.; Das, D. ToTTo: A Controlled Table-To-Text Generation Dataset. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 1173–1186.
9. Kukich, K. Design of a Knowledge-Based Report Generator. In Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics, Cambridge, MA, USA, 15–17 June 1983; pp. 145–150.
10. McKeown, K.R. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*; Cambridge University Press: Cambridge, MA, USA, 1985.
11. Gong, H.; Feng, X.; Qin, B.; Liu, T. Table-to-Text Generation with Effective Hierarchical Encoder on Three Dimensions (Row, Column and Time). In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the ninth International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 3143–3152.
12. Mei, H.; Bansal, M.; Walter, M.R. What to talk about and how? Selective Generation using LSTMs with Coarse-to-Fine Alignment. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 720–730.
13. Wang, Z.; Wang, X.; An, B.; Yu, D.; Chen, C. Towards Faithful Neural Table-to-Text Generation with Content-Matching Constraints. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 1072–1086.
14. Chen, W.; Su, Y.; Yan, X.; Wang, W.Y. KGPT: Knowledge-Grounded Pre-Training for Data-to-Text Generation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 8635–8648.
15. Chen, W.; Chen, J.; Su, Y.; Chen, Z.; Wang, W.Y. Logical Natural Language Generation from Open-Domain Tables. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7929–7942.
16. Nie, F.; Wang, J.; Yao, J.G.; Pan, R.; Lin, C.Y. Operation-guided Neural Networks for High Fidelity Data-To-Text Generation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October 2018; pp. 3879–3889.
17. Suadaa, L.H.; Kamigaito, H.; Funakoshi, K.; Okumura, M.; Takamura, H. Towards Table-to-Text Generation with Numerical Reasoning. 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), Online, 1–6 August 2021; pp. 1451–1465.
18. Chen, Z.; Chen, W.; Zha, H.; Zhou, X.; Zhang, Y.; Sundaresan, S.; Wang, W.Y. Logic2Text: High-Fidelity Natural Language Generation from Logical Forms. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online, 16–20 November 2020; pp. 2096–2111.
19. Zhang, N.; Ye, H.; Yang, J.; Deng, S.; Tan, C.; Chen, M.; Huang, S.; Huang, F.; Chen, H. LOGEN: Few-shot Logical Knowledge-Conditioned Text Generation with Self-training. *arXiv* **2021**. arXiv:2112.01404.
20. Chen, Z.; Eavani, H.; Chen, W.; Liu, Y.; Wang, W.Y. Few-Shot NLG with Pre-Trained Language Model. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 183–190.
21. Li, J.; Tang, T.; Zhao, W.X.; Wei, Z.; Yuan, N.J.; Wen, J.R. Few-shot Knowledge Graph-to-Text Generation with Pretrained Language Models. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online, 1–6 August 2021; pp. 1558–1568.
22. Jolly, S.; Zhang, Z.X.; Dengel, A.; Mou, L. Search and learn: Improving semantic coverage for data-to-text generation. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 22 February 2022; pp. 10858–10866.
23. Chang, E.; Shen, X.; Yeh, H.S.; Demberg, V. On Training Instance Selection for Few-Shot Neural 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 2: Short Papers), Online, 1–6 August 2021; pp. 8–13.
24. Dušek, O.; Howcroft, D.M.; Rieser, V. Semantic Noise Matters for Neural Natural Language Generation. In Proceedings of the 12th International Conference on Natural Language Generation, Tokyo, Japan, 28 October 2019; pp. 421–426.
25. Vinyals, O.; Fortunato, M.; Jaitly, N. Pointer Networks. In Proceedings of the Advances in Neural Information Processing Systems, Palais des Congrès de Montréal, QC, Canada, 7 December 2015; pp. 2692–2700.
26. Su, Y.; Meng, Z.; Baker, S.; Collier, N. Few-Shot Table-to-Text Generation with Prototype Memory. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2021, Punta Cana, Dominican Republic, 7–11 November 2021; pp. 910–917.

27. Zhao, W.; Liu, Y.; Wan, Y.; Yu, P. Attend, Memorize and Generate: Towards Faithful Table-to-Text Generation in Few Shots. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2021, Punta Cana, Dominican Republic, 7–11 November 2021; pp. 4106–4117.
28. Chang, E.; Shen, X.; Zhu, D.; Demberg, V.; Su, H. Neural Data-to-Text Generation with LM-based Text Augmentation. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, Online, 19–23 April 2021; pp. 758–768.
29. Chang, E.; Demberg, V.; Marin, A. Jointly Improving Language Understanding and Generation with Quality-Weighted Weak Supervision of Automatic Labeling. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Online, 19–23 April 2021; pp. 818–829.
30. Kasner, Z.; Dusek, O. Neural Pipeline for Zero-Shot Data-to-Text Generation. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; pp. 3914–3932.
31. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 7–9 August 2017; pp. 1126–1135.
32. Thrun, S.; Pratt, L. Learning to learn: Introduction and overview. In *Learning to Learn*; Springer: Boston, MA, USA 1998; pp. 3–17.
33. Volpi, R.; Larlus, D.; Rogez, G. Continual adaptation of visual representations via domain randomization and meta-learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Nashville, TN, USA, 20–25 June 2021; pp. 4443–4453.
34. Wang, C.; Pan, H.; Qiu, M.; Huang, J.; Yang, F.; Zhang, Y. Meta Distant Transfer Learning for Pre-trained Language Models. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, 7–11 November 2021; pp. 9742–9752.
35. Ren, M.; Zeng, W.; Yang, B.; Urtasun, R. Learning to reweight examples for robust deep learning. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 4334–4343.
36. Shu, J.; Xie, Q.; Yi, L.; Zhao, Q.; Zhou, S.; Xu, Z.; Meng, D. Meta-Weight-Net: Learning an Explicit Mapping For Sample Weighting. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8 December 2019; pp. 1919–1930.
37. Li, Z.; Nie, J.Y.; Wang, B.; Du, P.; Zhang, Y.; Zou, L.; Li, D. Meta-Learning for Neural Relation Classification with Distant Supervision. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual Event, Ireland, 19 October 2020; pp. 815–824.
38. Wu, L.; Xie, P.; Zhou, J.; Zhang, M.; Chunping, M.; Xu, G.; Zhang, M. Robust Self-Augmentation for Named Entity Recognition with Meta Reweighting. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Seattle, WA, USA, 10–15 July 2022; pp. 4049–4060.
39. Eshratifar, A.E.; Eigen, D.; Pedram, M. Gradient agreement as an optimization objective for meta-learning. In Proceedings of the second Workshop on Meta-Learning at NeurIPS 2018, Montreal, QC, Canada, 8 December 2018.
40. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.U.; Polosukhin, I. Attention is All you Need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4 December 2017; pp. 6000–6010.
41. Novikova, J.; Dušek, O.; Rieser, V. The E2E Dataset: New Challenges For End-to-End Generation. In Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrücken, Germany, 15–17 August 2017; pp. 201–206.
42. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. BLEU: A method for automatic evaluation of machine translation. In Proceedings of the Annual meeting on Association for Computational Linguistics, Philadelphia, PA, USA, 6 July 2002; pp. 311–318.
43. Doddington, G. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In Proceedings of the Second International Conference on Human Language Technology Research, San Francisco, CA, USA, 24 March 2002; pp. 138–145.
44. Banerjee, S.; Lavie, A. 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, Ann Arbor, MI, USA, 23 June 2005; pp. 65–72.
45. Lin, C.Y. Rouge: A package for automatic evaluation of summaries. In Proceedings of the Text Summarization Branches Out, Barcelona, Spain, 25–26 July 2004; pp. 74–81.
46. Vedantam, R.; Lawrence Zitnick, C.; Parikh, D. Cider: Consensus-based image description evaluation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Boston, MA, USA, 7–12 June 2015; pp. 4566–4575.
47. Ott, M.; Edunov, S.; Baevski, A.; Fan, A.; Gross, S.; Ng, N.; Grangier, D.; Auli, M. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), Minneapolis, Minnesota, 2–7 June 2019; pp. 48–53.
48. See, A.; Liu, P.J.; Manning, C.D. Get To The Point: Summarization with Pointer-Generator Networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July 2017; pp. 1073–1083.
49. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]

50. Gardent, C.; Shimorina, A.; Narayan, S.; Perez-Beltrachini, L. The WebNLG Challenge: Generating Text from RDF Data. In Proceedings of the tenth International Conference on Natural Language Generation, Santiago de Compostela, Spain, 4–7 September 2017; pp. 124–133.
51. Cho, K.; van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In Proceedings of the SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014; pp. 103–111.
52. Luong, T.; Pham, H.; Manning, C.D. Effective Approaches to Attention-based Neural Machine Translation. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 1412–1421.
53. Gu, J.; Lu, Z.; Li, H.; Li, V.O. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; pp. 1631–1640.
54. Fleiss, J.L. Measuring nominal scale agreement among many raters. *Psychol. Bull.* **1971**, *76*, 378. [CrossRef]