TWT: Table with Written Text for Controlled Data-to-Text Generation

Tongliang Li[1 2][*], **Lei Fang**[2], **Jian-Guang Lou**[2], and **Zhoujun Li**[1][†]
[1]State Key Lab of Software Development Environment, Beihang University, Beijing, China
[2]Microsoft Research, Beijing, China
{tonyliangli,lizj}@buaa.edu.cn, {leifa,jlou}@microsoft.com

Abstract

Large pre-trained neural models have recently shown remarkable progress in text generation. In this paper, we propose to generate text conditioned on the structured data (table) and a prefix (the written text) by leveraging the pre-trained models. We present a new data-to-text dataset, **T**able with **W**ritten **T**ext (TWT), by repurposing two existing datasets: ToTTo and TabFact. TWT contains both factual and logical statements that are faithful to the structured data, aiming to serve as a useful benchmark for controlled text generation. Compared with existing data-to-text task settings, TWT is more intuitive, the prefix (usually provided by the user) controls the topic of the generated text. Existing methods usually output hallucinated text that is not faithful on TWT. Therefore, we design a novel approach with table-aware attention visibility and copy mechanism over the table. Experimental results show that our approach outperforms state-of-the-art methods under both automatic and human evaluation metrics.

1 Introduction

Data-to-text refers to the task of generating a target textual description conditioned on the structured source data such as tables, graphs, and meaning representations. Reiter and Dale (1997) suggest that a natural language generation (NLG) system consists of content planning (what to say) and surface realization (how to say it). Recent deep neural network-based approaches do not explicitly model these stages and are trained in an end-to-end fashion using the popular encoder-decoder architecture (Sutskever et al., 2014) with the attention mechanism (Dzmitry et al., 2015; Lebret et al., 2016). They achieved promising results on existing data-to-text datasets, such as WebNLG (Gardent et al., 2017), E2ENLG (Novikova et al., 2017), WikiBio (Lebret et al., 2016), ROTOWIRE (Wiseman et al., 2017), ToTTo (Parikh et al., 2020), and LogicNLG (Chen et al., 2020a).

It should be noted that content planning is the key factor for data-to-text generation (Puduppully et al., 2019). Different users might interpret different parts of the structured data. This issue may not be severe for datasets (e.g. WebNLG (Gardent et al., 2017)) that require the generated text to cover all records. However, when the golden sentence only covers part of the records (e.g. WikiBio (Lebret et al., 2016)), end-to-end methods that do not explicitly address content planning may output open-ended targets, which leads to unreliable generated results, and places challenges in evaluation.

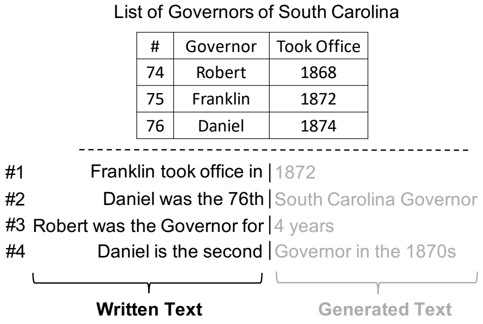
In NLG, one way to provide signals on what to generate is to add constraints to the model output, which falls in the task of controlled text generation (CTG). Most CTG tasks are conditioned on several key-value pairs of control factors such as tone, tense, length, and sentiment (Hu et al., 2017; Dong et al., 2017; Ficler and Goldberg, 2017). In data-to-text, Parikh et al. (2020) propose the dataset ToTTo to address content planning by highlighting some cells in the table, the highlighted cells provide strong guidance on what to generate. However, ToTTo lacks practical use, it would be difficult to have tables with highlighted cells or ask the users to highlight the cells in the real application.

One important application of NLG is to provide writing assistance such as next word prediction or text auto-completion. In this scenario, a natural content planning signal will be the written text provided by the user, which could be a word, a phrase, or an incomplete sentence. For the example shown in Figure 1, given the table, users might interpret different parts of the data with different prefixes. Text generation under this scenario requires inferring the user's intention on content planning based

[*]This work was done when the first author was an intern at Microsoft Research.

[†]Corresponding Author

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 1244–1254
November 7–11, 2021. ©2021 Association for Computational Linguistics


List of Governors of South Carolina

#	Governor	Took Office
74	Robert	1868
75	Franklin	1872
76	Daniel	1874

#1 Franklin took office in | 1872
#2 Daniel was the 76th | South Carolina Governor
#3 Robert was the Governor for | 4 years
#4 Daniel is the second | Governor in the 1870s

Figure 1: Data-to-text generation conditioned on the written text.

on the structured data and the written prefix.

To encourage the research in controlled data-to-text generation, we present a new dataset, **T**able with **W**ritten **T**ext (TWT), by repurposing two existing datasets: ToTTo (Parikh et al., 2020) and TabFact (Chen et al., 2019). See Section 3 for details about the dataset construction. TWT contains both factual and logical statements that are faithful to the structured data. Compared with other datasets, TWT is of practical use. The prefix controls the topic of the generated text, and the output model could assist in writing with structured data. Note that TWT differs from those datasets that provide only one golden sentence with no content planning signals.

To generate text faithful to the data, we design a novel approach that leverages large pre-trained models (Rothe et al., 2020) with table-aware attention visibility (based on the written text) and copy mechanism (Oriol et al., 2015; Gu et al., 2016) over the table. Experimental results show that our approach outperforms state-of-the-art methods under both automatic and human evaluation metrics, particularly in terms of faithfulness to the structured data. These results suggest that TWT could be a useful controllable data-to-text benchmark, and may help innovate models to provide intelligent assistance for writing with structured data.

2 Related Work

Data-to-Text aims to generate natural language from structured data, which has been widely studied recently. Most prior works focus on surface-level text generation in a specific domain or schema, such as ROBOCUP (Chen and Mooney, 2008), WEATHERGOV (Liang et al., 2009), E2ENLG (Novikova et al., 2017), and WebNLG (Gardent et al., 2017). These datasets expect the generated text to describe all the records from the data. WikiBio (Lebret et al., 2016) requires the target text to cover salient records with no explicit guidance on the generated topic. ToTTo (Parikh et al., 2020) guide the topic of the generated target with a set of highlighted table cells. Logic-NLG (Chen et al., 2020a) and Logic2Text (Chen et al., 2020b) address logical inference/generation in data-to-text. ROTOWIRE (Wiseman et al., 2017) and ToTTo (Parikh et al., 2020) also contain data that requires reasoning.

Many existing works tend to train neural models in an end-to-end fashion (Liu et al., 2018; Wiseman et al., 2017, 2018; Chen et al., 2020c). Recently, large pre-trained models (Rothe et al., 2020; Raffel et al., 2020; Lewis et al., 2020) have also achieved new state-of-the-art results on data-to-text tasks. Reiter and Dale (1997) suggest that an NLG system consists of content planning and surface realization. Parikh et al. (2020) propose ToTTo to control the topics of generated text with highlighted cells. Gong et al. (2020) brings the sense of numerical value comparison into content planning. Li and Wan (2018) propose to generate templates and then fill the slots, while (Iso et al., 2019) incorporate writers' information to generate text step-by-step. Gong et al. (2019) utilize hierarchical encoders with dual attention to consider both the table structure and history information. In NLG, controlled text generation is also a hot research topic. It considers controlling attributes, such as identity of the speaker (Li et al., 2016), sentiment (Dou et al., 2018), tense (Hu et al., 2017), politeness (Sennrich et al., 2016) and text length (Kikuchi et al., 2016). Our work could be considered as a middle-ground between data-to-text and controlled text generation and has more practical usage.

3 Task Definition and Dataset Construction

3.1 Task Definition

The task input is a tuple of table $\mathbb{T}$, metadata $\mathbb{M}$, and a written prefix X. The metadata $\mathbb{M}$ may include the table caption, the title of the section that contains the table, or other context around the table. The output target is denoted by Y, such that concatenating the prefix X and the output target Y results in a fluent sentence that is faithful to the table $\mathbb{T}$. The goal is to learn a data-to-text model conditioned on the written prefix, $P(Y|\mathbb{T}, \mathbb{M}, X)$.

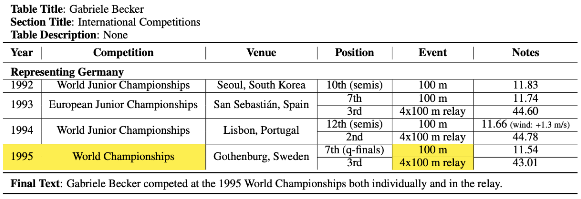
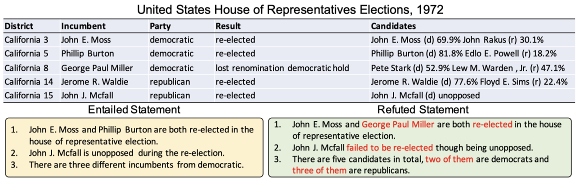


Table Title: Gabriele Becker
Section Title: International Competitions
Table Description: None

Year	Competition	Venue	Position	Event	Notes
Representing Germany					
1992	World Junior Championships	Seoul, South Korea	10th (semis)	100 m	11.83
1993	European Junior Championships	San Sebastián, Spain	7th	100 m	11.74
			3rd	4x100 m relay	44.60
1994	World Junior Championships	Lisbon, Portugal	12th (semis)	100 m	11.66 (wind: +1.3 m/s)
			2nd	4x100 m relay	44.78
1995	World Championships	Gothenburg, Sweden	7th (q-finals)	100 m	11.54
			3rd	4x100 m relay	43.01

Final Text: Gabriele Becker competed at the 1995 World Championships both individually and in the relay.

Figure 2: ToTTo dataset example (Parikh et al., 2020).



United States House of Representatives Elections, 1972

District	Incumbent	Party	Result	Candidates
California 3	John E. Moss	democratic	re-elected	John E. Moss (d) 69.9% John Rakus (r) 30.1%
California 5	Phillip Burton	democratic	re-elected	Phillip Burton (d) 81.8% Edlo E. Powell (r) 18.2%
California 8	George Paul Miller	democratic	lost renomination democratic hold	Pete Stark (d) 52.9% Lew M. Warden , Jr. (r) 47.1%
California 14	Jerome R. Waldie	republican	re-elected	Jerome R. Waldie (d) 77.6% Floyd E. Sims (r) 22.4%
California 15	John J. Mcfall	republican	re-elected	John J. Mcfall (d) unopposed

Entailed Statement

1. John E. Moss and Phillip Burton are both re-elected in the house of representative election.
2. John J. Mcfall is unopposed during the re-election.
3. There are three different incumbents from democratic.

Refuted Statement

1. John E. Moss and George Paul Miller are both re-elected in the house of representative election.
2. John J. Mcfall failed to be re-elected though being unopposed.
3. There are five candidates in total, two of them are democrats and three of them are republicans.

Figure 3: TabFact dataset example (Chen et al., 2019).

Property	ToTTo	TabFact
Number of prefix-target pairs	27,042	13,955
Average prefix length (tokens)	10.9	9.3
Average target length (tokens)	15.8	14.2
Rows per table (average/median)	32.8/16.0	10.9/10.0
Columns per table (average/median)	6.8/6.0	6.1/6.0

Table 1: TWT evaluation benchmark statistics.

3.2 Dataset Construction

Constructing a data-to-text dataset with clean targets is a significant challenge (Parikh et al., 2020), we therefore build TWT by repurposing two existing datasets: 1) ToTTo (Parikh et al., 2020), a large-scale controlled table-to-text generation dataset with highlighted cells and 2) TabFact (Chen et al., 2019), a table-based fact-checking dataset with rich logical annotated statements. As shown in Figure 2, in ToTTo, given the table, table metadata (such as the table title), and a set of highlighted cells, the goal is to produce the text that describes the highlighted cells. In TabFact, the input is a table with the caption and some statements (Figure 3), the task is to distinguish which statements are entailed or refuted. We use all annotated sentences from ToTTo and the entailed statements from TabFact as the clean targets. Chen et al. (2020a) address that data-to-text models should be able to generate text with logical inference over the data. Note that both ToTTo and TabFact contain text with logical inference. In total, we collected $128,268$ and $49,417$ table-sentence pairs from ToTTo and TabFact, respectively. After that, we resplit the table-sentence pairs to train/validation/test set as the TWT dataset. The size of the train/validation/test set for ToTTo source is $113,063/7,690/7,515$ and for TabFact is $39,678/5,009/4,730$.

Now, we could build the prefix and the golden target to generate by simulating the user writing process. An easy way to build prefix-target pairs is to break the sentence into two parts randomly, the first part will be the written prefix, and the second part is the target text to generate. However, the difficulty of generating correct target text on different breakpoints is not equal. Therefore, we build TWT evaluation benchmark with selected breakpoints in the sentence on the test set. These breakpoints are carefully selected such that the target contains either fact or logic derived from the table.

We employ a rule-based approach to choose the challenging breakpoints. We consider words or phrases that co-exist in the sentence and the table (or table metadata) as aligned facts. Following Chen et al. (2019), we identify the aligned facts based on the proportion of common words and word frequency of the longest common words between the text and each table cell or table metadata. For some text, we find that it contains numbers that do not exist in the table or table metadata (#3 and #4 in Figure 1). These numbers are usually logically inferred from the data. We consider these numbers as inferred numbers. The position to break the sentence will be the first starting token (excluded) of aligned facts and non-ordinal inferred numbers. For ordinal inferred numbers such as "first", "second" (#4 in Figure 1), the position will be the last token of the ordinal number (excluded). Once the positions to break the sentence are determined, we break the sentence at each position with the requirement that the prefix contains at least one aligned fact. Note that for sentences with multiple aligned facts or numbers, we will have multiple prefix-target pairs for one table-sentence pair. Table 1 shows the statistics of the obtained TWT evaluation benchmark.

4 Evaluation Metrics

For evaluation on TWT, we adopt the commonly used metrics in text generation, including BLEU score (Papineni et al., 2002), BLEURT (Sellam et al., 2020), and BERTScore (Zhang et al., 2020). Additionally, we introduce faithfulness metrics to measure the faithfulness of the generated text. Note that models trained on TWT might provide intelligent writing assistance, we also design several metrics specifically targeting this scenario.

4.1 Faithfulness Metrics

We propose two evaluation metrics to measure the faithfulness: fact coverage and the modified PARENT (Dhingra et al., 2019).

Fact Coverage is similar to the entity-centric metric (Liu et al., 2021), and the overall slot filling metric (Wang et al., 2018). Let $\mathbb{F}_g$ be the set of aligned facts of the golden target and the table data, and $\mathbb{F}_p$ be that for the generated target. Fact coverage is calculated as $|\mathbb{F}_p \cap \mathbb{F}_g|/|\mathbb{F}_g|$. Note that fact coverage of open-ended generated targets will be quite low. We use the same alignment method described in Section 3.2 to acquire $\mathbb{F}_g$ and $\mathbb{F}_p$.

PARENT (Dhingra et al., 2019) is a metric specifically designed for data-to-text evaluation that takes the input table into account. It computes smoothed n-gram precision and recall over both the generated target and the input table. Parikh et al. (2020) modifies this metric by computing the recall on the highlighted cells on ToTTo. Similarly, we calculate the recall on the set of aligned facts between the golden target and the data.

4.2 Text Prediction Metrics

In the scenario of providing writing assistance, whether the generated target can be accepted by the user depends on 1) whether the generated text matches the user's intention, and 2) how much writing effort can be saved. We design the following metrics targeting this scenario.

EM@N, the ratio of generated text whose words exactly match the first N words in the golden text.

Characters Saved, the number of matched characters between the generated and golden text. This metric measures how useful the model can help to save the writing efforts.

5 Methodology

With transformer-based structures, finetuning task-specific models with pre-trained parameters has achieved state-of-the-art results in text generation, achieving an astonishing level of fluency and coherence. Pre-trained models with a encoder-decoder structure such as BART (Lewis et al., 2020), BERT2BERT (Rothe et al., 2020), and T5 (Raffel et al., 2020) can be easily applied to data-to-text tasks. For example, on ToTTo, feeding the highlighted cells with row and column header as input and finetuned with BERT2BERT or T5 achieves relatively high performance (Parikh et al., 2020).

Figure 4 presents an overview of our model. We use a transformer-based encoder with additional positional (row/column) embeddings to encode table structure. We introduce structured encoder-decoder attention visibility based on the prefix to attend to the prefix-relevant sub-structure of the original table. For the decoder, we employ bi-directional attention for the prefix and uni-directional attention for the generated target as the decoder self-attention visibility. We also introduce the copy mechanism over the table data to assure the faithfulness of the generated target. Note that our model is based on the transformer encoder-decoder architecture (Rothe et al., 2020), both the encoder and the decoder are initialized with pre-trained parameters.

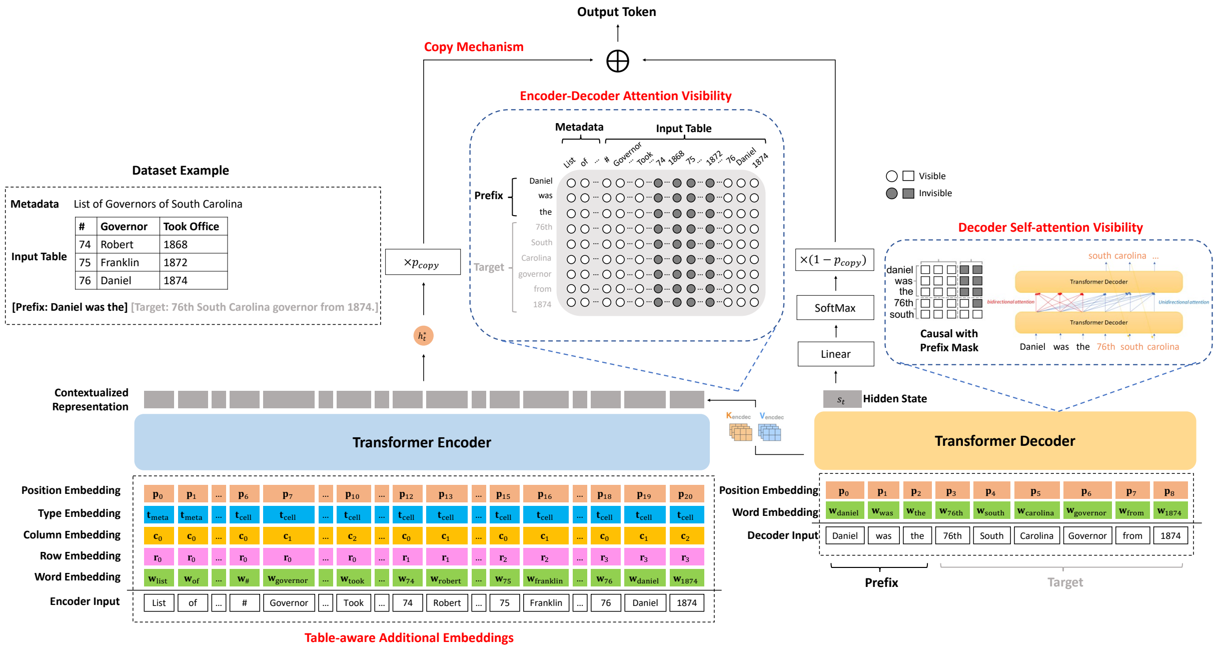
5.1 Table-aware Additional Embeddings

A common way to encode structured data with transformer is to create a linearized sequence of the data and treat the linearized sequence as text. For table linearization, similar to Yin et al. (2020), we use the template $h_c \mid h_r \mid v$ to represent each table cell, where h_c and h_r are column and row names of the cell v. Following Herzig et al. (2020) to represent the table structure, we add row embedding $\mathbf{r}$ and column embedding $\mathbf{c}$. We also use a type embedding $\mathbf{t}$ to represent the input type, where the type could be the table cell or different metadata types.

Given the input data, we first linearize the table row by row into a sequence of words and concatenate words of the metadata before the table words. The words are further tokenized with the WordPiece (Johnson et al., 2017) or SentencePiece (Kudo and Richardson, 2018) tokenizer. Let $\mathbf{p}$ be the positional embedding, $\mathbf{w}$ be the word embedding, and $\mathbf{e}$ denote the input representation, we have $\mathbf{e} = \mathbf{w} + \mathbf{r} + \mathbf{c} + \mathbf{t} + \mathbf{p}$.

5.2 Encoder-Decoder Attention Visibility

The prefix provides the content planning signals on the structured data. For example, in Figure 4, the prefix "Daniel was the" indicates that the following text is related to the row or column that "Daniel" belongs to with high probability. Therefore, we build a visibility matrix V based on the prefix as the encoder-decoder attention mask to explicitly model the visible row and column structure during decoding. $V_{i,j} = 1$ means that the token$_i$ (the encoder part) is visible to token$_j$ (the decoder part). We first extract the aligned facts for the prefix with



Metadata List of Governors of South Carolina

Input Table

#	Governor	Took Office
74	Robert	1868
75	Franklin	1872
76	Daniel	1874

[Prefix: Daniel was the] [Target: 76th South Carolina governor from 1874.]

Figure 4: Model overview.

the table records, and $V_{i,j} = 1$ if token$_i$ (the encoder part) belongs to the table metadata $\mathbb{M}$ or is from the same row/column of the aligned facts.

5.3 Decoder Self-attention Visibility

Typically, the encoder-decoder based models generate text starting from the beginning, and the decoder adopts a causal mask to force the state of each decoder time step s_{t_i} only attend to the state from the previous time steps, $s_{t|t \leq t_i}$, to avoid seeing tokens "from the future". We consider this type of attention as unidirectional. In our task, we have the input prefix as the written text. Tokens in the prefix should be visible to each other. Therefore, we adopt the causal with prefix mask: bidirectional attention mask is applied to the prefix, unidirectional attention is for decoding new tokens.

5.4 Copy Mechanism

To improve the faithfulness of the generated text, copying mechanism (Oriol et al., 2015; Gu et al., 2016) that copying from the data records is considered to be a promising solution (Li and Wan, 2018). Following (Chen et al., 2020c), on each decoding step t, we maintain a soft copy switch p_{copy} to choose between generating from the distribution over vocabulary, or copying from the input data with attention weights as the probability distribution:

$$p_{copy} = \sigma(w_x^T x_t + w_s^T s_t + w_{h_*}^T h_t^* + b)$$

where w_x, w_s, w_{h_*}, and b are learnable parameters, x_t is the decoder input, s_t is the output of the last decoder layer, σ is the sigmoid function, and h_t^* is the context vector, $h_t^* = \sum_i a_i^t h_i$, a_i^t is the encoder-decoder attention weight that masked with visibility introduced in Section 5.2.

Note that for the multi-head attention, we obtain p_{copy} by averaging that of all heads. Let $P_{vocab}(w)$ be the probability of generating token w, which is calculated through two linear layers with the concatenation of s_t and h_t^* as input (see See et al. (2017) for details), the final probability distribution over the extended vocabulary from the input data will be:

$$P(w) = (1 - p_{copy}) P_{vocab}(w) + p_{copy} \sum_{i:w_i=w} a_i^t$$

Copy mechanism is mainly proposed to handle out-of-vocabulary words (OOV) (Oriol et al., 2015; Gu et al., 2016). However, in our task, many of the table values are not OOV. The reason we employ the copy mechanism is to explicitly "teach" the model when and which fact to copy from the input data to improve faithfulness. We consider tokens of the aligned facts in the golden target as copied tokens, denoted by V_a. Following Chen et al. (2020c), we maximize the copy probability p_{copy} with an extra loss term at the copied tokens:

$$L = L_c + \lambda \sum_{w_j \in V_a} (1 - p_{copy}^j) \qquad (1)$$

Source	Model	BLEU	BLEURT	BERTScore	Writing Suggestion			Generation Faithfulness	
					EM@1 (%)	EM@2 (%)	Char Saved	Fact Coverage (%)	PARENT
ToTTo	T5	30.51	-0.46	0.34	36.79	26.37	11.70	33.48	8.04
	BERT2BERT	29.41	-0.40	0.36	32.68	22.49	13.13	30.66	8.19
	Ours (init from T5)	37.38	-0.27	0.45	50.24	37.62	14.65	46.68	11.58
	Ours (init from BERT)	33.47	-0.27	0.41	39.01	28.88	14.48	38.02	10.22
TabFact	T5	17.88	-0.70	-0.04	24.68	14.34	4.82	22.29	2.87
	BERT2BERT	15.33	-0.72	0.08	20.41	11.06	5.26	20.28	2.45
	Ours (init from T5)	24.18	-0.54	0.22	37.31	22.77	7.86	36.13	6.90
	Ours (init from BERT)	18.69	-0.66	0.18	23.80	13.77	5.56	23.49	2.98

Table 2: Experimental results on the TWT evaluation benchmark. Our models adopt the "Causal with Prefix" decoder mask pattern, which uses bidirectional attention mask for prefix, and unidirectional attention mask for decoding new tokens (see Section 5.3 for details).

Model	Source	Averaged Score
T5	ToTTo	1.48
BERT2BERT		1.49
Ours (init from T5)		1.91
Ours (init from BERT)		1.77
T5	TabFact	1.36
BERT2BERT		1.25
Ours (init from T5)		1.87
Ours (init from BERT)		1.32

Table 3: Human evaluation scores. Our model uses the causal with prefix mask for the decoder self-attention.

where L_c is the original loss between the model's output and the golden target, w_j is the target token at position j. λ is a hyper-parameter representing the weight for the copy.

6 Experiments [1]

Following Parikh et al. (2020) on selecting the baselines on ToTTo, we exam the following state-of-the-art text generation approaches on TWT.

- **BERT2BERT** (Rothe et al., 2020): A Transformer encoder-decoder model where the encoder and decoder are both initialized with BERT (Devlin et al., 2019).

- **T5** (Raffel et al., 2020): A pre-trained text-to-text using the transformer framework. T5 achieved state-of-the-art results on many text generation benchmarks, including ToTTo.

Note that for baseline models, the input is the metadata concatenated with the table flattened row by row, with no additional table-aware embeddings introduced in Section 5.1.

6.1 Setup

We build the prefix-target pairs for training and validation by randomly selecting two prefixes of each table-sentence pair from the TWT train/validation set. The number of prefix-target pairs built for training/validation is $226,126/15,380$ from the ToTTo source and $79,356/10,018$ from the TabFact source. The trained model is then tested on the TWT evaluation benchmark.

For our approach, we initialize the parameters of encoder and decoder with two variants: BERT (Devlin et al., 2019) following BERT2BERT (Rothe et al., 2020) and T5 (Raffel et al., 2020), with the remaining parameters initialized randomly. When initialized with BERT, encoder and decoder do not share parameters. The learning rate is $5e$–5. We use the linear learning rate scheduler with Adam optimizer (Kingma and Ba, 2015), and use beam search with the beam size of 4 during decoding. When initialized with T5, following (Raffel et al., 2020), we employ a constant learning rate of $1e-3$ with AdaFactor optimizer (Shazeer and Stern, 2018). Decoding is conducted via greedy search. For other settings (including the baselines), the batch size is 56, and the maximum number of input and output tokens are 512 and 128, respectively. Tokens that exceed the maximum length will be truncated. We tune the hyper-parameter λ of the copy weight (Equation 1) and set it to 0.4, which achieves the best overall performance. We train both baselines and our approach with 8 NVIDIA Tesla V100 32G GPUs. The best checkpoint is chosen based on the fact coverage metric on the validation set.

6.2 Experimental Results

Table 2 shows the comparison between our approach and the baselines. We observe that: 1) our approach outperforms the baseline methods on all metrics, and 2) on both data sources, our approach

[1] Our code, data, and model are publicly available at https://aka.ms/emnlp_twt.

Source	Model	BLEU	BLEURT	BERTScore	Writing Suggestion			Generation Faithfulness	
					EM@1 (%)	EM@2 (%)	Char Saved	Fact Coverage (%)	PARENT
ToTTo	Ours (init from T5)	37.38	-0.27	0.45	50.24	37.62	14.65	46.68	11.58
	- w/o causal with prefix	36.41	-0.28	0.44	50.94	38.10	14.54	45.97	11.22
	Ours (init from BERT)	33.47	-0.27	0.41	39.01	28.88	14.48	38.02	10.22
	- w/o col/row embeddings	33.31	-0.27	0.41	39.09	28.78	14.55	38.23	10.34
	- w/o enc-dec attn visibility	30.82	-0.38	0.38	34.25	24.48	12.72	31.46	8.30
	- w/o copy mechanism	33.04	-0.28	0.41	38.37	28.19	14.47	37.60	10.15
	- w/o causal with prefix	31.52	-0.35	0.38	37.33	26.71	13.25	34.55	9.05
TabFact	Ours (init from T5)	24.18	-0.54	0.22	37.31	22.77	7.86	36.13	6.90
	- w/o causal with prefix	24.13	-0.55	0.20	35.34	22.09	7.46	33.56	5.45
	Ours (init from BERT)	18.69	-0.66	0.18	23.80	13.77	5.56	23.49	2.98
	- w/o causal with prefix	16.45	-0.70	0.09	22.92	12.88	5.59	22.05	2.52

Table 4: Ablation studies, "w/o causal with prefix" means we replace it with the causal mask (unidirectional).

initialized with T5 achieves the best performance.

The improvements on the faithfulness metrics are more significant. The results of the writing suggestion metrics also demonstrate that our approach could help reduce writing efforts with structured data in real applications.

6.3 Ablation Study[2]

We conduct ablation studies to investigate the model designs of our approach: 1) the table structure-aware additional embeddings, 2) the structured encoder-decoder attention visibility, 3) the copy mechanism, and 4) the “causal with prefix” decoding mask pattern. The results of different variants are listed in Table 4.

The overall performance drops when we employ unidirectional decoding mask on both sources when initialized with BERT or T5, suggesting that it’s effective to employ the bidirectional attention mask to the prefix. On the ToTTo source data, it can be seen that, when the parameters are initialized with BERT, the overall performance of all metrics drops without the encoder-decoder attention visibility (enc-dec attn visibility) or the copy mechanism. The results also suggest that introducing the table structure-aware column and row embeddings doesn’t show improvements (the results are comparable). We leave this as our future work to further study how to represent tables in transformer-based model structures. The overall results demonstrate that these designs are effective to achieve improved performance.

6.4 Human Evaluation

In our task, some correct and faithful generated text might be different from the golden targets, which results in low performance using the above automatic evaluation metrics. The predictions of our models in Figure 5 Case #2 could be an example of this type. To further evaluate the faithfulness of the generated target, we randomly select 200 samples from the test set and ask the annotators to judge the predictions in terms of factual and logical correctness. We score 3/2/1 to each generated text indicating the facts or logic are all/partially/not correct.

Table 3 shows the averaged scores of human evaluation. Compared with baselines, our approach generates more faithful text on data from the ToTTo source, and when initialized with T5, our approach achieves the best overall scores on data from both sources. We also find that the performance is rather poor when the golden target contains logical inference over the data. We leave this as our future work.

6.5 Case Study

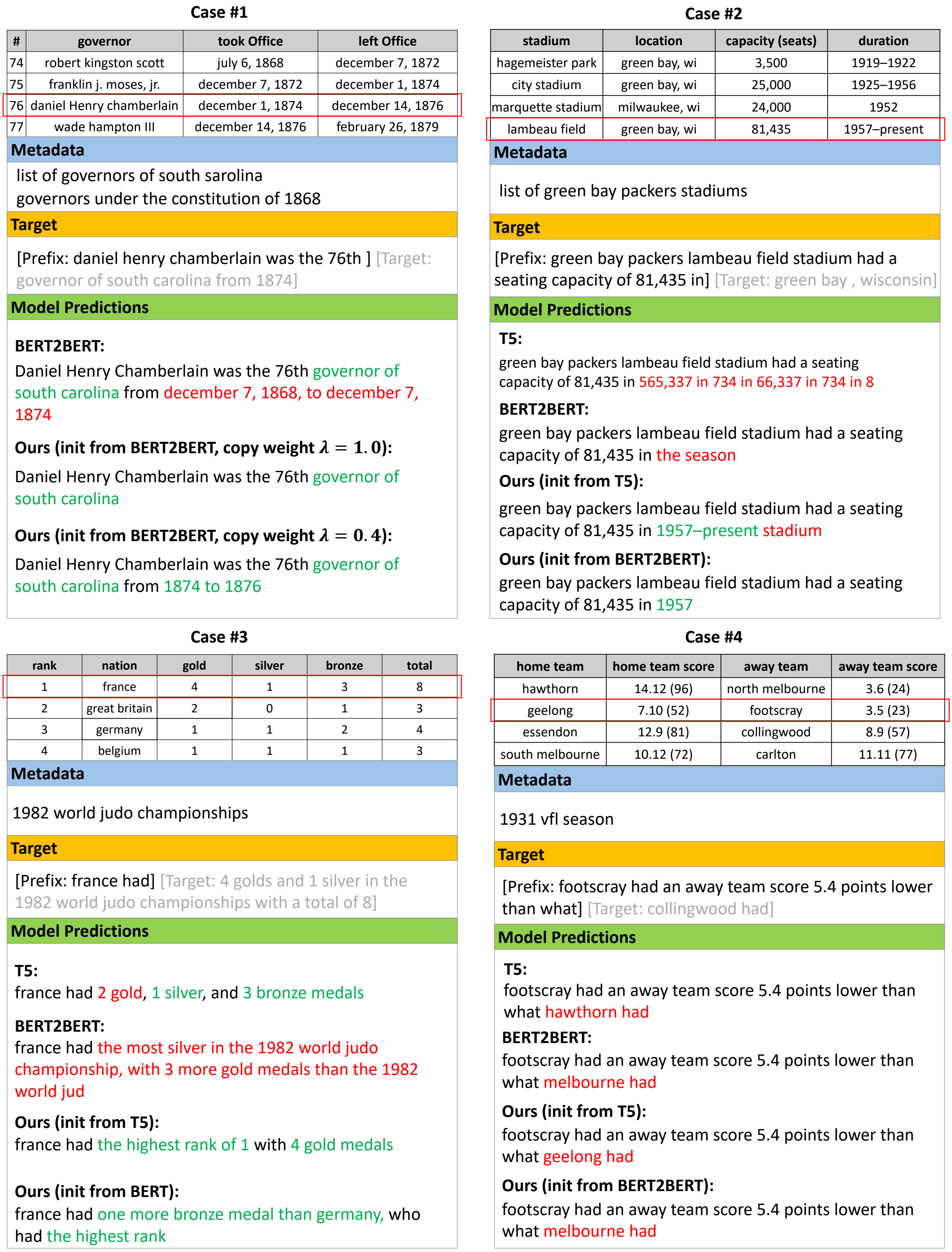
Figure 5 shows the generated text of several cases for baselines and our approach.

Case #1 shows how the copy mechanism affects the generated text. Increasing the value of λ makes the model “reluctant” to generate new text beyond the table content, and we find that the larger the value of λ is, the shorter the output text will be. λ balances between quality (faithfulness) and diversity. Note that "to 1876" in Case #1 is faithful to the table, which is not included in the target.

In Case #2, all baseline models generate unfaithful results while our models generate faithful ones, the output of our approach shall be considered as correct even though it’s different from the golden target. This case demonstrates that, with encoder-decoder attention visibility, our model could focus on a specific sub-structure of the table to generate more faithful results.

In Case #3, the prefix is not sufficient to guide

[2] Due to limited computation resources, we do ablation studies mainly for our approach initialized with BERT on the ToTTo source.



Case #1

#	governor	took Office	left Office
74	robert kingston scott	july 6, 1868	december 7, 1872
75	franklin j. moses, jr.	december 7, 1872	december 1, 1874
76	daniel Henry chamberlain	december 1, 1874	december 14, 1876
77	wade hampton III	december 14, 1876	february 26, 1879

Metadata

list of governors of south sarolina
governors under the constitution of 1868

Target

[Prefix: daniel henry chamberlain was the 76th] [Target: governor of south carolina from 1874]

Model Predictions

BERT2BERT:
Daniel Henry Chamberlain was the 76th governor of south carolina from december 7, 1868, to december 7, 1874

Ours (init from BERT2BERT, copy weight $\lambda = 1.0$):
Daniel Henry Chamberlain was the 76th governor of south carolina

Ours (init from BERT2BERT, copy weight $\lambda = 0.4$):
Daniel Henry Chamberlain was the 76th governor of south carolina from 1874 to 1876

Case #2

stadium	location	capacity (seats)	duration
hagemeister park	green bay, wi	3,500	1919–1922
city stadium	green bay, wi	25,000	1925–1956
marquette stadium	milwaukee, wi	24,000	1952
lambeau field	green bay, wi	81,435	1957–present

Metadata

list of green bay packers stadiums

Target

[Prefix: green bay packers lambeau field stadium had a seating capacity of 81,435 in] [Target: green bay , wisconsin]

Model Predictions

T5:
green bay packers lambeau field stadium had a seating capacity of 81,435 in 565,337 in 734 in 66,337 in 734 in 8

BERT2BERT:
green bay packers lambeau field stadium had a seating capacity of 81,435 in the season

Ours (init from T5):
green bay packers lambeau field stadium had a seating capacity of 81,435 in 1957–present stadium

Ours (init from BERT2BERT):
green bay packers lambeau field stadium had a seating capacity of 81,435 in 1957

Case #3

rank	nation	gold	silver	bronze	total
1	france	4	1	3	8
2	great britain	2	0	1	3
3	germany	1	1	2	4
4	belgium	1	1	1	3

Metadata

1982 world judo championships

Target

[Prefix: france had] [Target: 4 golds and 1 silver in the 1982 world judo championships with a total of 8]

Model Predictions

T5:
france had 2 gold, 1 silver, and 3 bronze medals

BERT2BERT:
france had the most silver in the 1982 world judo championship, with 3 more gold medals than the 1982 world jud

Ours (init from T5):
france had the highest rank of 1 with 4 gold medals

Ours (init from BERT):
france had one more bronze medal than germany, who had the highest rank

Case #4

home team	home team score	away team	away team score
hawthorn	14.12 (96)	north melbourne	3.6 (24)
geelong	7.10 (52)	footscray	3.5 (23)
essendon	12.9 (81)	collingwood	8.9 (57)
south melbourne	10.12 (72)	carlton	11.11 (77)

Metadata

1931 vfl season

Target

[Prefix: footscray had an away team score 5.4 points lower than what] [Target: collingwood had]

Model Predictions

T5:
footscray had an away team score 5.4 points lower than what hawthorn had

BERT2BERT:
footscray had an away team score 5.4 points lower than what melbourne had

Ours (init from T5):
footscray had an away team score 5.4 points lower than what geelong had

Ours (init from BERT2BERT):
footscray had an away team score 5.4 points lower than what melbourne had

Figure 5: Case studies. Text segments colored in green means the content is faithful to the data, and those colored in red are unfaithful content.

the model to generate factual or logical content. Our model still outperforms the baseline models, the model attempts to generate text which involves logical inference. Our model does not explicitly model logic, the reason might be that the logic here is relatively simple, which does not require algebra calculation over the numbers.

Case #4 shows that when the logic involved is complex, all models including ours fail to generate the correct result. We leave generating text with logical inference over the data as our future work.

7 Task Challenges

Logical Inference. Text generation with logical inference over the data is challenging in our task. For example, the golden target of Case #4 in Figure 5 requires calculation over the numerical values in the table.

Choosing between Fact and Logic. In TWT, the golden target contains both factual and logical text. The model shall be capable of choosing what type of content to generate. For example, in Case #3 of Figure 5, the target sentence is factual while the model attempts to generate logical text, which leads to low evaluation results, though the predicted text is correct.

Evaluation metrics. A good text generation model shall be capable of generating diverse and faithful content, which is not limited to generating results close to the provided target. Case #2 is an example of this type. The results of Ours (init from BERT2BERT) shall be considered correct. Even for the evaluation metrics, we find that these metrics usually are not consistent. For example, a high BLEU score does not necessarily mean that the fact coverage or PARENT metric is high.

8 Conclusion

In this paper, we propose **T**able with **W**ritten **T**ext (TWT), a new controlled data-to-text generation dataset. For this task, we design a novel approach with table-aware attention visibility and copy mechanism over the table. Experimental results show that our approach could generate faithful text over state-of-the-art pre-trained models under both automatic and human evaluation. For future work, we will focus on generating text with logical inference on TWT.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Grant Nos.U1636211, 61672081, 61370126) and the Fund of the State Key Laboratory of Software Development Environment (Grant No.SKLSDE-2021ZX-18).

References

David L. Chen and Raymond J. Mooney. 2008. Learning to sportscast: A test of grounded language acquisition. In *Proceedings of the 25th International Conference on Machine Learning*, pages 128–135.

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.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyangu Li., Xiyou Zhou, and William Yang Wang. 2019. Tabfact: A large-scale dataset for table-based fact verification. In *Proceedings of the 7th International Conference on Learning Representations*.

Zhiyu Chen, Wenhu Chen, Hanwen Zha, Xiyou Zhou, Yunkai Zhang, Sairam Sundaresan, and William Yang Wang. 2020b. Logic2Text: High-fidelity natural language generation from logical forms. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2096–2111.

Zhiyu Chen, Harini Eavani, Wenhu Chen, Yinyin Liu, and William Yang Wang. 2020c. Few-shot NLG with pre-trained language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 183–190.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

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.

Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. 2017. Learning to generate product reviews from attributes. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 623–632.

Longxu Dou, Guanghui Qin, Jinpeng Wang, Jin-Ge Yao, and Chin-Yew Lin. 2018. Data2Text studio: Automated text generation from structured data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 13–18.

Bahdanau Dzmitry, Kyung Hyun Cho, and Bengio Yoshua. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, pages 94–104.

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.

Heng Gong, Wei Bi, Xiaocheng Feng, Bing Qin, Xiaojiang Liu, and Ting Liu. 2020. Enhancing content planning for table-to-text generation with data understanding and verification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2905–2914.

Heng Gong, Xiaocheng Feng, Bing Qin, and Ting Liu. 2019. 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 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3143–3152.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. 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)*, pages 1631–1640.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1587–1596.

Hayate Iso, Yui Uehara, Tatsuya Ishigaki, Hiroshi Noji, Eiji Aramaki, Ichiro Kobayashi, Yusuke Miyao, Naoaki Okazaki, and Hiroya Takamura. 2019. Learning to select, track, and generate for data-to-text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2102–2113.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, pages 339–351.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1328–1338.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

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.

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.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 994–1003.

Liunian Li and Xiaojun Wan. 2018. Point precisely: Towards ensuring the precision of data in generated texts using delayed copy mechanism. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1044–1055.

Percy Liang, Michael Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 91–99.

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018. Table-to-text generation by structure-aware seq2seq learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 4881–4888.

Tianyu Liu, Xin Zheng, Baobao Chang, and Zhifang Sui. 2021. Towards faithfulness in open domain table-to-text generation from an entity-centric view. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*, pages 13415–13423.

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.

Vinyals Oriol, Fortunato Meire, and Jaitly Navdeep. 2015. Pointer networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, pages 2692—-2700.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

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.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, pages 6908–6915.

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*, pages 1–67.

Ehud Reiter and R. Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering*, pages 57–87.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, pages 264–280.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. 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)*, pages 1073–1083.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40.

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

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, pages 3104–3112.

Qingyun Wang, Xiaoman Pan, Lifu Huang, Boliang Zhang, Zhiying Jiang, Heng Ji, and Kevin Knight. 2018. Describing a knowledge base. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 10–21.

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.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2018. Learning neural templates for text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3174–3187.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with bert. In *Proceedings of the 8th International Conference on Learning Representations*.