Towards Table-to-Text Generation with Pretrained Language Model: A Table Structure Understanding and Text Deliberating Approach

Miao Chen[§][*], Xinjiang Lu[¶][⊠], Tong Xu[§], Yanyan Li[¶], Jingbo Zhou[¶], Dejing Dou[¶], Hui Xiong[†][‡]
[¶]BIL, Baidu Research, [§]University of Science and Technology of China
[†]Hong Kong University of Science and Technology (Guangzhou)
[‡]Guangzhou HKUST Fok Ying Tung Research Institute
`cmer@mail.ustc.edu.cn`, `{luxinjiang,liyanyanliyanyan,zhoujingbo}@baidu.com`,
`tongxu@ustc.edu.cn`, `doudejing@baidu.com`, `xionghui@ust.hk`

Abstract

Although remarkable progress on the neural table-to-text methods has been made, the generalization issues hinder the applicability of these models due to the limited source tables. Large-scale pretrained language models sound like a promising solution to tackle such issues. However, how to effectively bridge the gap between the structured table and the text input by fully leveraging table information to fuel the pretrained model is still not well explored. Besides, another challenge of integrating the deliberation mechanism into the text-to-text pretrained model for solving the table-to-text task remains seldom studied. In this paper, to implement the table-to-text generation with pretrained language model, we propose a table structure understanding and text deliberating approach, namely TASD. To be specific, we devise a three-layered multi-head attention network to realize the table-structure-aware text generation model with the help of the pretrained language model. Furthermore, a multi-pass decoder framework is adopted to enhance the capability of polishing generated text for table descriptions. The empirical studies, as well as human evaluation, on two public datasets, validate that our approach can generate faithful and fluent descriptive texts for different types of tables.

1 Introduction

The task of learning to generate natural language descriptions from non-linguistic input, which is referred to as data-to-text, is important for many applications, such as weather forecast generation (Mei et al., 2016), sports news writing (Wiseman et al., 2017), biography writing (Lebret et al., 2016), market comments writing (Murakami et al., 2017) and automatic question-answering (Li et al., 2021b). The input data can be in various forms for data-to-text though, here we focus on the text generation task that takes the table as input.

Inspired by neural machine translation models, previous studies on table-to-text tasks mainly adopt traditional seq2seq methods to generate table descriptions (Lebret et al., 2016; Wiseman et al., 2017; Liu et al., 2018; Gong et al., 2019b; Wang et al., 2020; Li et al., 2021a). Despite generating text with high fluency, lacking numerous source tables leads to lower generalizability of the table-to-text model. Recent progress in the pretrained language model (Devlin et al., 2019; Radford et al., 2019) shows remarkable performance in solving natural language processing tasks. The model pretrained on large-scale data possesses rich knowledge, which inspires us with the potential for solving generalization issues of the text generation task.

To exploit the expressive power of the pretrained model for the table-to-text task, it is necessary to serialize the input table effectively. Several works have put efforts to bridge this gap, such as serializing the table into a token sequence (Zhang et al., 2020; Suadaa et al., 2021; Xing and Wan, 2021), or introducing an extra task to control the table representation (Gong et al., 2020). However, none of these leveraged the table structure information effectively. Furthermore, the text-to-text pretrained model decodes and generates a sequence in a one-pass forward process, which means it cannot perceive the future words in advance on the target side. Recently, the deliberation mechanism (Niehues et al., 2016; Geng et al., 2018) implemented by the multi-pass decoder is proposed to tackle this problem. However, how to adapt this approach for text-to-text pretraining, which can be further applied to the table-to-text task, is another challenge.

To this end, we propose a table structure understanding and text deliberating approach, namely TASD, to solve the table-to-text task with the pretrained language model enhanced by the deliberation mechanism. Specifically, we first serialize

---

[*] This work was done when the first author was an intern at Baidu Research under the supervision of the second author.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 8199–8210
December 7-11, 2022 ©2022 Association for Computational Linguistics

the table input with customized templates which do not acquire the target cells to be labeled. Then, we employ the multi-head attention in a hierarchical way to learn the table representation that is aware of table structure and apply it to guide the fine-tuning of the text-to-text pretrained model. Afterward, we adopt the multi-pass decoder to realize text deliberation. More specifically, we treat the above table-structure-aware fine-tuned model as the first-pass decoder and adopt another pretrained model as the second-pass decoder to further polish the descriptive text. In the second-pass decoding phase, the table representation can be conveniently leveraged as the "original text" in the text deliberation mechanism. The main contributions of this work can be summarized as follows:

- We propose a novel table-to-text generation approach (i.e., TASD) to assimilating the complete table information with the help of table structure distillation, the pretrained language model, and the text deliberation.
- We devise a table-structure-aware text generation model (TASATG) via the hierarchical multi-head attention network, which can realize the content selection automatically. And we develop an effective text deliberation method dedicated to the table-to-text task.
- Extensive experiments conducted on two different datasets demonstrate that TASD outperforms comparable baselines in terms of various metrics.

2 Related Work

2.1 Table-to-Text Generation

Encouraged by the success of seq2seq methods in machine translation and text summarization, researchers proposed to formulate the input table as a sequence of records (Lebret et al., 2016; Wiseman et al., 2017), and further improve the performance of table-to-text methods based on seq2seq by modeling table representation (Liu et al., 2018; Gong et al., 2019a). Introducing auxiliary tasks to enrich the table representation (Tian et al., 2019; Li et al., 2021a) is another promising paradigm to address the table-to-text problem. Moreover, there have been studies focusing on how to disaggregate the table-to-text pipeline effectively to generate more faithful and fluent text, e.g. leveraging content selection and planning (Puduppully et al., 2019; Trisedya et al., 2020; Bai et al., 2021), combining autoregressive and non-autoregressive methods (Wang et al., 2021). In addition, recent Transformers were also applied to solve the table-to-text task (Gong et al., 2019b; Wang et al., 2020; Obeid and Hoque, 2020). However, current table-to-text methods may fail to tackle the overfitting problem aroused by the lack of diversity in small datasets.

Fine-tuning the model pretrained in a large corpus and adapting to a specific task is an effective approach to tackling the generation issues disturbed by small data and large parameters (Radford et al., 2019). (Kale and Rastogi, 2020) explored the feasibility of applying the text-to-text pretrained model to the table-to-text task, (Gong et al., 2020) applied multi-task learning to solve the table-to-text task with pretrained language model, and (Suadaa et al., 2021) leveraged pretrained language model for fact inference in numerical table contents. However, these approaches seldom perceived and integrated the complete table information into the fine-tuning of the pretrained model. A table-to-text pretrained model (Xing and Wan, 2021) was proposed though, the large and diversified table corpus is often unavailable. In addition, recent works on fact verification taking tabular as input (Yin et al., 2020; Dong and Smith, 2021) have suggested the effectiveness of the table-structure-aware pretrained model.

2.2 Text Deliberation

The encoder-decoder framework has been widely applied to neural machine translation, while the subsequent words are often invisible on the target side when decoding a sequence. To alleviate this, researchers proposed to decode and refine the output sequence in multiple passes, like human cognitive behavior when polishing an article. Studies have been made on text deliberation, such as the solution with two separate stages (i.e., generating and polishing) (Niehues et al., 2016), combining two separate stages as one framework (Xia et al., 2017), and deliberating generated text in multiple passes adaptively via reinforcement learning (Geng et al., 2018) or customized evaluating architecture (Li and Yao, 2021). To the best of our knowledge, we are the first to apply the deliberation mechanism to the table-to-text problem.

3 Preliminaries

3.1 Problem Formulation

Our table-to-text problem takes a table as input, and we formulate a table as a sequence of records: $T = \{\tau_{1,1}, \tau_{1,2}, \cdots, \tau_{i,j}, \cdots, \tau_{m,n}\}$, where m and n denote the number of rows and columns of T, re-

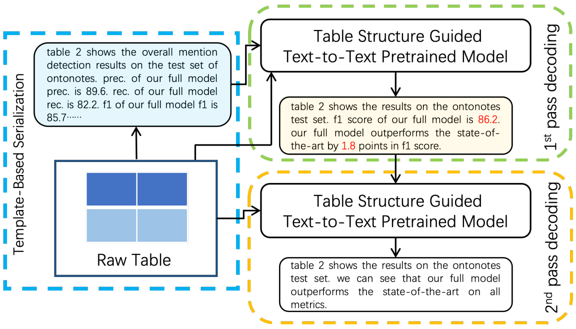

Template-Based Serialization
Raw Table
Table Structure Guided Text-to-Text Pretrained Model
1st pass decoding
Table Structure Guided Text-to-Text Pretrained Model
2nd pass decoding

table 2 shows the overall mention detection results on the test set of ontonotes. prec. of our full model prec. is 89.6. rec. of our full model rec. is 82.2. f1 of our full model f1 is 85.7……

table 2 shows the results on the ontonotes test set. f1 score of our full model is 86.2. our full model outperforms the state-of-the-art by 1.8 points in f1 score.

table 2 shows the results on the ontonotes test set. we can see that our full model outperforms the state-of-the-art on all metrics.

Figure 1: The framework overview of TASD.

spectively. Then, we aim to generate a document Y containing words $Y = y_1 y_2 \cdots y_l$ that can describe the content of T precisely, where l is the document length. Formally, given a table T, the table-to-text model is excepted to generate a descriptive document Y in an auto-regressive way

$$y_i = \arg\max P(y_i \mid T, y_1 y_2 \cdots y_{i-1}; \Theta),\ i = 1, \cdots, l$$

where Θ is the set of model parameters.

3.2 Data

NumericNLG Dataset. The numericNLG dataset was released by (Suadaa et al., 2021). In this dataset, the tables demonstrate experimental results in research papers, thus, most of the table contents are numerical values. We use this dataset to evaluate the accuracy and smoothness of the generated descriptions for the table with numerical content. In particular, for each table of numericNLG, `<table_id>` acts as the pronoun of the table, and `<caption>` is the descriptive text of the table. Moreover, for each cell of a table, there are `<metric>`, (`row and column`) `<header>`, and `<value>` as different views of a cell.

Totto Dataset. The Totto dataset (Parikh et al., 2020) is an open-domain table-to-text dataset collected from Wikipedia. The table contents are mainly in text form. The metadata of the Totto dataset includes `<page_title>`, `<section_title>` and `<section_text>`. In detail, each cell of a table has corresponding `<header>` and `<value>`. Unlike numericNLG, textual content in our Totto dataset accounts for 62.4%, which can evaluate the text generation effectiveness for the tables with textual records.

4 Methodology

In this section, we introduce the proposed framework in detail. As shown in Fig. 1, our framework mainly consists of three components, i.e., *template-based table serialization*, *table-structure-aware fine-tuning*, and *text deliberation*. Specifically, we first produce a sequence describing the table contents with customized templates. The templates we adopted do not require the target cells to be labeled. Then, to generate informative text, we adopt full table representation learning to guide the description generation, such that the outcome text is capable of emphasizing and delineating the facts in the table from a macroscopic perspective. Finally, we employ and adapt the multi-pass decoder to our data-to-text problem, which can further fine-tune the generated table description. Technical details for all three modules will be introduced separately in the following subsections.

4.1 Template-based Table Serialization

To well harness the expressive power of the text-to-text pretrained model for the input table, it is necessary to serialize the raw table first. The template-based representation offers us a simple yet effective linearization approach to generating descriptive texts which can reflect the facts in a table without yielding an intractable downstream model.

In particular, the templates we adopted in this work are devised to mention all the available facts in the table without knowing the emphasized cells in advance, which is different from (Suadaa et al., 2021). The template for describing facts consists of two parts:

1. The title or descriptive text that comes with the table.
2. A series of expressions, in which each one describes the content of a cell.

More specifically, for the numericNLG dataset, we apply the following template:

> `<table_id>` shows `<caption>`. `<metric`$_{1,1}$`>` of `<header`$_{1,1}$`>` is `<value`$_{1,1}$`>`, $\cdots$, `<metric`$_{i,j}$`>` of `<header`$_{i,j}$`>` is `<value`$_{i,j}$`>`, $\cdots$.

For the Totto dataset, we apply another template:

> As `<page_title>` `<section_title>`, `<section_text>`. `<header`$_{1,1}$`>` is `<value`$_{1,1}$`>`, $\cdots$, `<header`$_{i,j}$`>` is `<value`$_{i,j}$`>`, $\cdots$.

The second part of the template enumerates all the cells in the table. This preliminary table representation, denoted by T_S, covers all the available facts in a raw table. Note that, the templates we adopt may encounter the content selection problem. In table-to-text applications, target cells in the input table are often not highlighted and the generated table description should emphasize certain cells.

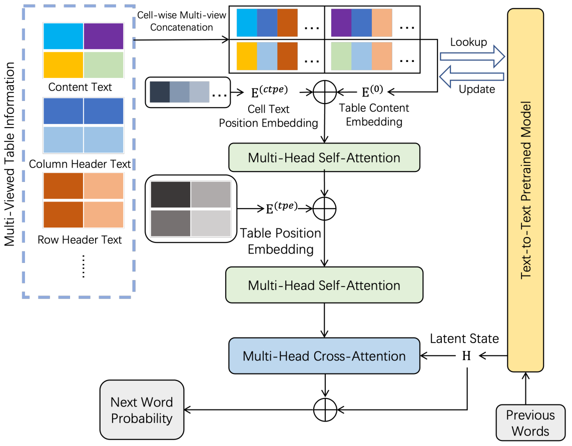


Figure 2: The architecture of table-structure-aware text generation model (i.e., TASATG).

4.2 Table-Structure-Aware Text Generation

A text-to-text pretrained model can take the large-scale corpus as input to possess vast knowledge and generate texts in an unsupervised way so that it has been widely applied to text-generation tasks. When handling a specific text generation task, it is effective to fine-tune the pretrained model on new data. However, for the table-to-text task, some hidden information, like table structure, is most likely to be overlooked, though the drafted T_S mentions all the available facts in the table. Thus, we propose to exploit table structure information to guide fine-tuning of the text-to-text pretrained model.

As shown in Fig. 2, we first encode the table content in a multi-view fashion. To be specific, given a cell $\tau_{i,j}$ in a table T, it can be viewed from different perspectives, such as the value of $\tau_{i,j}$, the row header of $\tau_{i,j}$, and the column header of $\tau_{i,j}$, etc. Then, we treat the k-th view of $\tau_{i,j}$ as a token sequence which is denoted by $x_{i,j}^{(k)}$. Afterward, we pad $x_{i,j}^{(k)}$ with placeholders (if necessary) and concatenate these token sequences as follows:

$$\mathbf{x}_{i,j} = x_{i,j}^{(1)} \circledast x_{i,j}^{(2)} \circledast \cdots, \tag{1}$$

where $\circledast$ denotes the concatenation operator, and the multi-viewed representation of a table T is denoted as $\mathbf{X} = [\mathbf{x}_{1,1}, \cdots, \mathbf{x}_{i,j}, \cdots, \mathbf{x}_{m,n}]$. Each token of $\mathbf{x}_{i,j}^{(k)}$ can be encoded as a d-dimensional embedding by looking up the text-to-text pretrained model and updated accordingly when fine-tuning the pretrained model. In this way, we can obtain the semantic representation of table T, which is denoted by $\mathrm{E}^{(0)} \in \mathbb{R}^{m \times n \times s \times d}$, where s is the length of concatenated sequence $\mathbf{x}_{i,j}$.

To realize TASATG for table-to-text, we propose to employ multi-head attention (Vaswani et al., 2017) to guide fine-tuning of the text-to-text pretrained model. In particular, we adopt three multi-head attention (MHA) layers to interactively extract the information in the table in a hierarchical way. Specifically, the MHA layer is defined as:

$$Q_i = QW_i^Q,\ K_i = KW_i^K,\ V_i = VW_i^V$$

$$\text{head}_{\,i} = \text{Attention}\,(Q_i, K_i, V_i) = \text{softmax}\left(\frac{Q_i K_i^\top}{\sqrt{d}}\right) V_i,$$

$$\text{MHA}(Q, K, V) = [\,\text{head}_{\,1}, \cdots, \text{head}_{\,h}]\, W^O,$$

where Q, K, V represent the query, key and value in the attention mechanism, respectively.

As illustrated in Fig. 2, in the first MHA layer, we add a cell text position embedding ($\mathrm{E}^{(ctpe)} \in \mathbb{R}^{s \times d}$) to each cell of the aforementioned $\mathrm{E}^{(0)}$, and feed it to the multi-head attention to implement cell text self-attention,

$$\begin{aligned}
\widetilde{\mathrm{E}_0} &= \mathrm{E}^{(0)} \oplus \mathrm{E}^{(ctpe)}, \\
\mathrm{E}^{(1)} &= \text{MHA}(\widetilde{\mathrm{E}^{(0)}}, \widetilde{\mathrm{E}^{(0)}}, \widetilde{\mathrm{E}^{(0)}}), \\
\mathrm{E}^{(1)} &= \frac{1}{s} \sum_{i=1}^{s} (\mathrm{E}^{(1)}[:, :, i, :])\,,
\end{aligned} \tag{2}$$

where $\oplus$ denotes the element-wise addition operation. Consequently, $\mathrm{E}^{(1)} \in \mathbb{R}^{m \times n \times d}$ can be deemed as an initial aggregated table representation. Next, in the second MHA layer, we add a table position embedding ($\mathrm{E}^{(tpe)} \in \mathbb{R}^{m \times n \times d}$) to $\mathrm{E}^{(1)}$ to implement table structure self-attention,

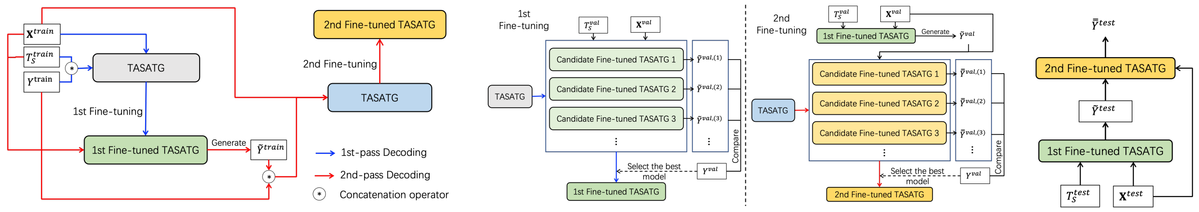
$$\begin{aligned}
\widetilde{\mathrm{E}^{(1)}} &= \mathrm{E}^{(1)} \oplus \mathrm{E}^{(tpe)}, \\
\mathrm{E}^{(2)} &= \text{MHA}(\widetilde{\mathrm{E}^{(1)}}, \widetilde{\mathrm{E}^{(1)}}, \widetilde{\mathrm{E}^{(1)}}).
\end{aligned} \tag{3}$$

$\mathrm{E}^{(2)} \in \mathbb{R}^{m \times n \times d}$ is the table-structure-aware representation. Moreover, in the third MHA layer, we apply a multi-head cross-attention to take the hidden state of the text-to-text pretrained model (denoted by $\mathrm{H} \in \mathbb{R}^{s \times d}$) as the attention query, such that we can focus on the important cells of the table,

$$\widetilde{\mathrm{H}} = \text{MHA}(\mathrm{H}, \mathrm{E}^{(2)}, \mathrm{E}^{(2)}) \oplus \mathrm{H}. \tag{4}$$

This new hidden state $\widetilde{\mathrm{H}}$ guided by the table representation will replace the original hidden state H in the text-to-text pretrained model to generate the probability of the next word.

Note that, the cross attention weights on different table cells based on the previous words can realize the content selection automatically. In addition, we implement the text-to-text pretrained model with GPT2 (Radford et al., 2019), which adopts a decoder-only Transformer architecture.


$\mathbf{X}^{train}$
T_S^{train}
Y^{train}
TASATG
1st Fine-tuning
1st Fine-tuned TASATG
Generate
$\tilde{Y}^{train}$
2nd Fine-tuned TASATG
2nd Fine-tuning
TASATG
1st-pass Decoding
2nd-pass Decoding
Concatenation operator
1st Fine-tuning
T_S^{val}
$\mathbf{X}^{val}$
TASATG
Candidate Fine-tuned TASATG 1
Candidate Fine-tuned TASATG 2
Candidate Fine-tuned TASATG 3
$\tilde{Y}^{val,(1)}$
$\tilde{Y}^{val,(2)}$
$\tilde{Y}^{val,(3)}$
Compare
Select the best model
Y^{val}
1st Fine-tuned TASATG
2nd Fine-tuning
T_S^{val}
$\mathbf{X}^{val}$
1st Fine-tuned TASATG
Generate
$\tilde{Y}^{val}$
TASATG
Candidate Fine-tuned TASATG 1
Candidate Fine-tuned TASATG 2
Candidate Fine-tuned TASATG 3
$\bar{\bar{Y}}^{val,(1)}$
$\bar{\bar{Y}}^{val,(2)}$
$\bar{\bar{Y}}^{val,(3)}$
Compare
Select the best model
Y^{val}
2nd Fine-tuned TASATG
$\bar{\bar{Y}}^{test}$
2nd Fine-tuned TASATG
$\tilde{Y}^{test}$
1st Fine-tuned TASATG
T_S^{test}
$\mathbf{X}^{test}$

(a) Training.

(b) First and second fine-tuning of TASATG with validation data.

(c) Testing.

Figure 3: Training, validation and testing procedures of the proposed TASD approach.

4.3 Text Deliberation

The encoder-decoder framework applied in many sequence generation tasks often adopts a one-pass process while decoding a sequence. Though efficient, the one-pass decoder cannot perceive future context for further text deliberation. Multi-pass decoder extends the capability of generating more refined text by exploring global information in the sequence (Niehues et al., 2016; Xia et al., 2017).

For the text-to-text pretrained model, due to the huge amount of parameters of the pretrained language model, it is unwise to directly combine the models in different passes. A common solution is to concatenate the original serialized table content and the text generated in the previous pass to fine-tune the pretrained model in the next-pass decoding. However, in this way, the length of input text probably exceeds the limit of the text-to-text pretrained model, and the time complexity is too high.

To effectively implement the fine-tuning of the text-to-text pretrained model in multiple passes, as shown in Figs. 3a and 3b, we take the table representation as the "original text" and feed the text generated in the first-pass fine-tuning plus the table representation to the second-pass fine-tuning. Note that, as shown in Fig. 3a, we separately fine-tune the table-to-text generation task and the text-to-text deliberation task with two independent TASATG models, and each of them takes a text-to-text pretrained model as the backbone.

5 Experiments

5.1 Experimental Settings

Data. We conducted experiments on the aforementioned datasets, i.e., numericNLG and Totto. The statistics of the numericNLG dataset can be found in (Suadaa et al., 2021). Besides, the size of the original Totto dataset is 120K, which is much larger than the numericNLG dataset. To evaluate different methods for table-to-text with comparable data size, for the Totto dataset, we filtered out the tables with fewer rows and columns, i.e., #rows < 8 and #columns < 8, such that the filtered Totto dataset contains 1.8K tables. Then, we randomly selected 1.2K[1] tables to generate the new Totto dataset.

Evaluation Metrics. We calculated BLEU (from gram-1 to gram-4) (Papineni et al., 2002), ROUGE-L (Lin, 2004) and METEOR (Denkowski and Lavie, 2014) to evaluate the quality of the generated text. The BLEU-n with a small value of n measures the accuracy of the word level, and the BLEU-n with a large n can measure the fluency of the sentence. The ROUGE-L measures the recall rate based on the longest common sequence between source and target texts. The METEOR is based on the harmonic mean of unigram precision and recall, with recall weighted higher than precision. These metrics are widely used to measure the accuracy and fluency of the generated sentence.

Baselines. We compare TASD with the following baselines.

- **Template-based Table Serialization**. We use the template designed for table serialization as a baseline. Note that, the token sequence generated by the template-based method is denoted as T_S.
- **Pointer Generator** (See et al., 2017). This is a seq2seq model with the attention and copy mechanism. We take T_S as input for the pointer generator model.
- **TRM**. We implemented a simplified version of the proposed TASD that omits the possessed knowledge in the pretrained language model and removes text deliberation for focusing on table representation modeling, namely TRM. In particular, TRM adopts the architecture of GPT2 but initializes the parameters randomly and trains 100 epochs at most for fine-tuning. Besides, TRM takes T_S plus the table structure representation as input and is fed with T_S in the inference phase.

---

[1] The size of numericNLG data is 1.3K.

Table 1: Performance comparisons of the automatic evaluation on the numericNLG dataset.

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L
Template-based Method	10.28	5.52	2.83	1.14	11.31	11.49
Pointer Generator	$5.10_{\pm 0.59}$	$2.71_{\pm 0.19}$	$1.16_{\pm 0.17}$	$0.56_{\pm 0.04}$	$7.82_{\pm 0.15}$	$15.21_{\pm 0.14}$
TRM	$14.16_{\pm 0.97}$	$6.05_{\pm 0.50}$	$2.11_{\pm 0.13}$	$0.80_{\pm 0.12}$	$9.72_{\pm 0.94}$	$12.72_{\pm 0.80}$
Fine-tuned GPT2	$16.13_{\pm 0.56}$	$9.02_{\pm 0.31}$	$4.68_{\pm 0.22}$	$2.20_{\pm 0.22}$	$10.14_{\pm 0.32}$	$17.48_{\pm 0.36}$
TableGPT	$18.69_{\pm 0.39}$	$8.21_{\pm 0.24}$	$3.31_{\pm 0.19}$	$1.51_{\pm 0.14}$	$11.06_{\pm 0.18}$	$16.90_{\pm 0.27}$
TASD $_{w/o\ \text{TAS}}$	$18.20_{\pm 2.40}$	$9.74_{\pm 1.01}$	$4.38_{\pm 0.31}$	$1.98_{\pm 0.39}$	$10.64_{\pm 0.86}$	$19.29_{\pm 1.77}$
TASD $_{w/o\ \text{D}}$	$18.02_{\pm 0.50}$	$10.06_{\pm 0.25}$	$\mathbf{5.20}_{\pm 0.13}$	$\mathbf{2.47}_{\pm 0.20}$	$10.99_{\pm 0.29}$	$18.57_{\pm 0.27}$
TASD $_{w/o\ 1^{\text{st}}\text{-TAS}}$	$20.07_{\pm 1.94}$	$10.35_{\pm 0.69}$	$4.67_{\pm 0.35}$	$2.05_{\pm 0.34}$	$11.52_{\pm 0.80}$	$20.10_{\pm 0.62}$
TASD	$\mathbf{21.81}_{\pm 1.13}$	$\mathbf{11.03}_{\pm 0.11}$	$4.92_{\pm 0.22}$	$2.15_{\pm 0.39}$	$\mathbf{11.87}_{\pm 0.40}$	$\mathbf{20.40}_{\pm 0.80}$

Table 2: Performance comparisons of the automatic evaluation on the Totto dataset.

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L
Template-based Method	0.84	0.43	0.23	0.09	4.59	1.51
Pointer Generator	$11.34_{\pm 1.57}$	$2.05_{\pm 0.83}$	$0.45_{\pm 0.27}$	$0.35_{\pm 0.13}$	$5.38_{\pm 0.78}$	$14.46_{\pm 1.46}$
TRM	$10.21_{\pm 1.79}$	$3.44_{\pm 0.88}$	$1.21_{\pm 0.48}$	$0.54_{\pm 0.25}$	$9.30_{\pm 1.16}$	$11.52_{\pm 2.03}$
Fine-tuned GPT2	$9.53_{\pm 0.51}$	$3.65_{\pm 0.34}$	$1.18_{\pm 0.37}$	$0.40_{\pm 0.26}$	$9.89_{\pm 0.39}$	$10.69_{\pm 0.27}$
TableGPT	$6.80_{\pm 0.26}$	$3.51_{\pm 0.22}$	$1.33_{\pm 0.21}$	$0.76_{\pm 0.12}$	$11.10_{\pm 0.42}$	$11.73_{\pm 0.44}$
TASD $_{w/o\ \text{TAS}}$	$13.70_{\pm 0.90}$	$4.44_{\pm 0.69}$	$1.28_{\pm 0.47}$	$0.65_{\pm 0.35}$	$10.79_{\pm 0.83}$	$14.47_{\pm 1.11}$
TASD $_{w/o\ \text{D}}$	$10.03_{\pm 0.39}$	$4.42_{\pm 0.29}$	$1.64_{\pm 0.36}$	$0.71_{\pm 0.38}$	$10.29_{\pm 0.49}$	$10.67_{\pm 0.34}$
TASD $_{w/o\ 1^{\text{st}}\text{-TAS}}$	$13.90_{\pm 0.60}$	$5.07_{\pm 0.61}$	$1.68_{\pm 0.52}$	$\mathbf{0.79}_{\pm 0.25}$	$10.98_{\pm 0.40}$	$14.88_{\pm 0.71}$
TASD	$\mathbf{14.19}_{\pm 1.08}$	$\mathbf{5.17}_{\pm 0.38}$	$\mathbf{1.71}_{\pm 0.32}$	$0.78_{\pm 0.21}$	$\mathbf{11.65}_{\pm 0.71}$	$\mathbf{14.96}_{\pm 1.10}$

- **Fine-tuned GPT2** (Radford et al., 2019). We take the concatenation of T_S and Y as the input for fine-tuning. In the inference phase, we only feed T_S to the model to generate Y starting after the last token of T_S.
- **TableGPT** (Gong et al., 2020). TableGPT is a state-of-the-art table-to-text method. To improve the text fidelity and exploit the structural information at the same time, TableGPT employs a multi-task learning paradigm consisting of two auxiliary tasks, that is, one task reconstructs the table structure from representations of GPT2, and the other aligns the tables and the information in the generated text.

Implementation Details. The split settings for training, validation and, testing were 1084:136:135 [2] for the numericNLG dataset and 960:120:120 for the Totto dataset, respectively. Regarding automatic evaluation, all results of deep models were obtained by conducting experiments on a Linux machine with Nvidia A100 GPU, and the averaged results of 5 runs were reported. Besides, an Adam optimizer was utilized (with an initial learning rate of 3e-5) for GPT2 fine-tuning, and the training was iterated in 20 epochs at most. A beam search algorithm was adopted when decoding a sequence and the beam width was set to 5 [3].

5.2 Automatic Evaluation

The comparisons of automatic evaluation results between TASD and other baselines can be found in Tables 1 and 2. In general, TASD outperforms the baselines for all the metrics on two datasets. In particular, compared to the reported best result of all the baselines, TASD achieves improvements of 3.12 for BLEU-1 (18.69 → 21.81), 2.01 for BLEU-2 (9.02 → 11.03), 0.24 for BLEU-3 (4.68 → 4.92), 0.56 for METEOR (11.31 → 11.87), and 2.92 for ROUGE-L (17.48 → 20.40) on the numericNLG dataset, and 2.85 for BLEU-1 (11.34 → 14.19), 1.52 for BLEU-2 (3.65 → 5.17), 0.38 for BLEU-3 (1.33 → 1.71), 0.02 for BLEU-4 (0.76 → 0.78), 0.55 for METEOR (11.10 → 11.65), and 0.50 for ROUGE-L (14.46 → 14.96) on the Totto dataset.

---

[2] This setting follows the experiments of (Suadaa et al., 2021).

[3] Our implementation is available at `https://github.com/ramber1836/TASD`.

In other words, for different types of source tables, TASD generates better descriptive texts w.r.t. accuracy at the word level, recall of the sequence, and fluency of sentences.

Besides, we have the following observations: **1)** The template-based method performs much better on the numericNLG dataset compared to the Totto dataset, since the referenced table descriptions in numericNLG were collected from scientific papers, however, the table summaries in the Totto dataset are more diverse. **2)** In the Totto dadaset, the pointer generator model tends to cover more words in descriptive text and generate more fluent sentences than the template-based method, as the contents in source tables of the Totto dataset are mostly linguistic. This can also explain why the pointer generator performs worse than the template-based method on the numericNLG dataset w.r.t. BLEU and METEOR. **3)** Fine-tuned GPT2 can generate more faithful and fluent text than other baselines (refer to Tables 1 and 2) most of the time, which validates the effectiveness of the pretrained language model. **4)** In general, TableGPT performs better, and even the best, among all the baselines. In the numericNLG dataset, the headers of the input tables (a.k.a. the attributes of records for TableGPT) are more diverse, which may explain why the performance of TableGPT is not promising as expected on the numericNLG dataset. **5)** TRM can generate comparable, or even better descriptive text as fined-tuned GPT2, which further suggests the effectiveness of table structure understanding.

5.3 Ablation Analysis

Moreover, to verify the effectiveness of different modules, we compare TASD with its variants.

- After generating text with fine-tuned GPT2, we fed the generated text concatenated with T_S to another fine-tuned GPT2 to realize the second-pass decoder without table structure representation.
- We implemented TASD without deliberating on the outcome text, which means that we realized TASATG based on GPT2 in a one-pass forward process.
- TASD $_{w/o\ \text{1st-TAS}}$. We removed table structure modeling in the first-pass decoding from TASD, which was implemented by taking the fine-tuned GPT2 as the first-pass decoder and the table-structure-aware fine-tuned GPT2 as the second-pass decoder.

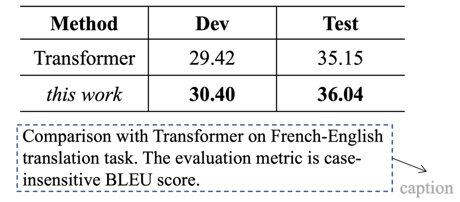
As can be seen in Tables 1 and 2, TASD $_{w/o\ \text{TAS}}$ performs worse than TASD under all metrics, since the table structure modeling can benefit the fine-tuning of GPT2. This can also be validated by comparing fine-tuned GPT2 to TASD $_{w/o\ \text{D}}$. Besides, the effectiveness of deliberating text can be proven by comparing TASD $_{w/o\ \text{D}}$ to TASD (this can also be validated by comparing fine-tuned GPT2 to TASD $_{w/o\ \text{TAS}}$). While text deliberation may harm sentence fluency as depicted by the results of these methods w.r.t. BLEU-3 & 4 in Table 1. In addition, TASD $_{w/o\ \text{1st-TAS}}$ outperforms TASD $_{w/o\ \text{TAS}}$ under all metrics suggesting that taking the table representation as the “original text” in the deliberation mechanism is also effective.

5.4 Qualitative Analysis

Figs. 4(a) and (b) show two selected source tables and corresponding descriptive texts (i.e., caption and section_text) in numericNLG and Totto datasets. Fig. 4(c) demonstrates the generated descriptions by different methods. The text that correctly reflects the facts of the source table is in green, the erroneous text is in red, and the confusing text is in blue. We can see that, there are many grammatical errors in the text produced by the pointer generator. Fine-tuned GPT2 tends to repeat phrases and sentences due to the limited knowledge about the input table, which can also explain why the fine-tuned GPT2 can obtain a false high score in BLEU-n as n grows. Thanks to the semantic knowledge brought by pretraining, fine-tuned GPT2 can generate more natural descriptions, in which, however, perplexing factual errors exist. Compared to fine-tuned GPT2, the description generated by TASD is more relevant to the table contents. Since the target cells are not known in advance, the generated text may miss the emphasized points described in the reference. The text generated by TableGPT is also fluent, though counterfactual descriptions may exist.

5.5 Human Evaluation

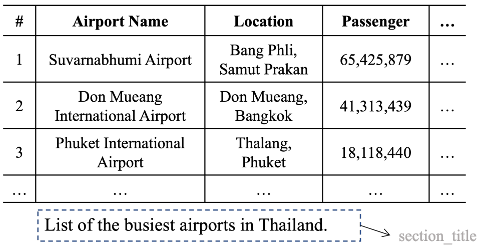
We randomly selected 30 samples from the test set in numericNLG and Totto datasets, respectively, and invited 10 volunteers to evaluate the quality of the outcome text by considering three criteria, i.e., grammar, coherence & concise, and factual perspective (correct and relevant). Each criterion has scores of five degrees, ranging from 1 (the worst) to 5 (the best). The averaged scores were reported in Table 3, which show that TASD can generate more



Method	Dev	Test
Transformer	29.42	35.15
this work	**30.40**	**36.04**

Comparison with Transformer on French-English translation task. The evaluation metric is case-insensitive BLEU score.

(a) Case A: a sample in the numericNLG dataset

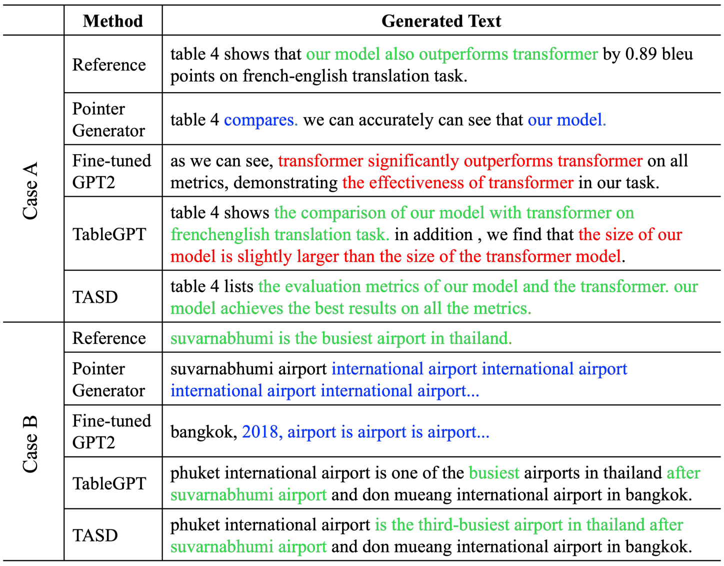


#	Airport Name	Location	Passenger	…
1	Suvarnabhumi Airport	Bang Phli, Samut Prakan	65,425,879	…
2	Don Mueang International Airport	Don Mueang, Bangkok	41,313,439	…
3	Phuket International Airport	Thalang, Phuket	18,118,440	…
…	…	…	…	…

List of the busiest airports in Thailand.

section_title

(b) Case B: a sample in the Totto dataset



	Method	Generated Text
Case A	Reference	table 4 shows that our model also outperforms transformer by 0.89 bleu points on french-english translation task.
	Pointer Generator	table 4 compares. we can accurately can see that our model.
	Fine-tuned GPT2	as we can see, transformer significantly outperforms transformer on all metrics, demonstrating the effectiveness of transformer in our task.
	TableGPT	table 4 shows the comparison of our model with transformer on frenchenglish translation task. in addition , we find that the size of our model is slightly larger than the size of the transformer model.
	TASD	table 4 lists the evaluation metrics of our model and the transformer. our model achieves the best results on all the metrics.
Case B	Reference	suvarnabhumi is the busiest airport in thailand.
	Pointer Generator	suvarnabhumi airport international airport international airport international airport international airport...
	Fine-tuned GPT2	bangkok, 2018, airport is airport is airport...
	TableGPT	phuket international airport is one of the busiest airports in thailand after suvarnabhumi airport and don mueang international airport in bangkok.
	TASD	phuket international airport is the third-busiest airport in thailand after suvarnabhumi airport and don mueang international airport in bangkok.

(c) Generated texts for case A and case B

Figure 4: Two examples of the generated table descriptions.

Table 3: Result of Human Evaluation

Dataset	Method	Grammar	Coherence & Concise	Factual perspective
numericNLG	Pointer Generator	$3.16_{\pm0.99}$	$2.73_{\pm1.20}$	$1.54_{\pm0.69}$
	Fine-tuned GPT2	$3.42_{\pm0.56}$	$3.11_{\pm0.58}$	$2.51_{\pm0.45}$
	TASD $_{w/o\ D}$	$3.72_{\pm0.61}$	$3.48_{\pm0.55}$	$2.82_{\pm0.45}$
	TASD	$\mathbf{4.17}_{\pm0.72}$	$\mathbf{3.98}_{\pm0.64}$	$\mathbf{3.15}_{\pm0.73}$
Totto	Pointer Generator	$2.03_{\pm0.71}$	$1.89_{\pm0.82}$	$1.56_{\pm0.55}$
	Fine-tuned GPT2	$2.60_{\pm0.55}$	$2.36_{\pm0.64}$	$1.85_{\pm0.46}$
	TASD $_{w/o\ D}$	$2.63_{\pm0.52}$	$2.46_{\pm0.60}$	$1.89_{\pm0.46}$
	TASD	$\mathbf{3.4}_{\pm0.66}$	$\mathbf{3.18}_{\pm0.70}$	$\mathbf{2.25}_{\pm0.69}$

readable and coherent texts, and describe more correct facts. Moreover, the pretrained models consistently achieve better scores than the pointer generator on grammar and coherence because of the expressive power learned from the large-scale corpus. In the Totto dataset, the improvement of the table structure modeling is smaller than that of the polishing mechanism, which is consistent with the automatic evaluation results in Table 2.

6 Discussion

In our work, we devised a two-pass decoder framework dedicated to the table-to-text task with the help of the table-structure-aware text generation model (i.e., TASATG). However, the effectiveness of the text deliberation for the table-to-text task should be further explored and integrated into the table-structure-aware modeling in a more harmonic manner. To discuss the limitation of the text deliberation of TASD, we additionally developed a table content reconstruction loss and integrate it into TASD in a multi-task learning fashion.

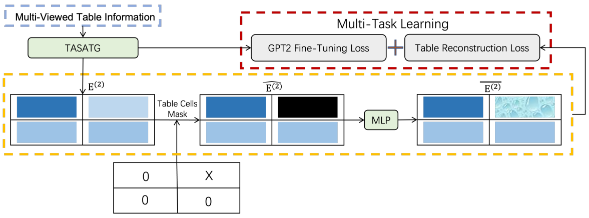


Figure 5: Table reconstruction for table-structure-aware modeling enhancement.

Specifically, given the table-structure-aware embedding $E^{(2)}$ generated with Eq. (3), we randomly mask certain cells of the input table and yield a partially corrupted embedding of the input table, denoted by $\widehat{E^{(2)}}$. Then, a two-layer MLP (i.e., multi-layer perceptron) is adopted to restore the table-structure-aware embedding. Afterward, an MSE (i.e., mean square error) loss is adopted to measure the effectiveness of table reconstruction and further integrated into the TASD framework in the multi-task learning paradigm. The process of table reconstruction is demonstrated in Fig. 5.

We carried out a series of experiments to evaluate the performance of TASD w/ and w/o the help of table reconstruction loss (i.e., TRLoss) on numericNLG and Totto datasets in terms of BLEU-n (1 to 4), METEOR, and ROUGE-L. The results can be found in Tables 4 and 5.

Table 4: The performances of TASD w/ and w/o the table reconstruction on the numericNLG dataset.

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L
TASD $_{w/o\ \text{D}}$	$18.02_{\pm 0.50}$	$10.06_{\pm 0.25}$	$5.20_{\pm 0.13}$	$2.47_{\pm 0.20}$	$10.99_{\pm 0.29}$	$18.57_{\pm 0.27}$
TASD $_{w/o\ \text{D}\ w/\ \text{TRLoss}}$	$20.56_{\pm 0.25}$	**11.57**$_{\pm 0.21}$	**5.90**$_{\pm 0.23}$	**2.98**$_{\pm 0.17}$	$12.00_{\pm 0.48}$	**20.50**$_{\pm 0.39}$
TASD $_{w/\ \text{TRLoss}}$	$19.29_{\pm 0.38}$	$10.12_{\pm 0.24}$	$5.32_{\pm 0.25}$	$2.62_{\pm 0.22}$	**12.18**$_{\pm 0.90}$	$18.95_{\pm 0.69}$
TASD $_{w/\ \text{TRLoss in 1st pass}}$	$18.23_{\pm 0.68}$	$9.39_{\pm 0.52}$	$4.64_{\pm 0.26}$	$2.36_{\pm 0.24}$	$11.51_{\pm 0.78}$	$18.13_{\pm 0.45}$
TASD $_{w/\ \text{TRLoss in 2nd pass}}$	$19.38_{\pm 2.21}$	$10.33_{\pm 1.34}$	$5.11_{\pm 0.73}$	$2.40_{\pm 0.38}$	$11.35_{\pm 0.92}$	$18.69_{\pm 1.05}$
TASD	**21.81**$_{\pm 1.13}$	$11.03_{\pm 0.11}$	$4.92_{\pm 0.22}$	$2.15_{\pm 0.39}$	$11.87_{\pm 0.40}$	$20.40_{\pm 0.80}$

Table 5: The performances of TASD w/ and w/o the table reconstruction on the Totto dataset.

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L
TASD $_{w/o\ \text{D}}$	$10.03_{\pm 0.39}$	$4.42_{\pm 0.29}$	$1.64_{\pm 0.36}$	$0.71_{\pm 0.38}$	$10.29_{\pm 0.49}$	$10.67_{\pm 0.34}$
TASD $_{w/o\text{D}\ w/\ \text{TRLoss}}$	$9.94_{\pm 0.43}$	$4.35_{\pm 0.31}$	$1.63_{\pm 0.31}$	$0.75_{\pm 0.13}$	$10.37_{\pm 0.22}$	$10.62_{\pm 0.60}$
TASD $_{w/\ \text{TRLoss}}$	**14.57**$_{\pm 0.87}$	$5.22_{\pm 0.42}$	$1.70_{\pm 0.49}$	**0.89**$_{\pm 0.38}$	**11.79**$_{\pm 0.77}$	**15.28**$_{\pm 0.86}$
TASD $_{w/\ \text{TRLoss in 1st pass}}$	$14.00_{\pm 0.82}$	**5.31**$_{\pm 0.27}$	**1.72**$_{\pm 0.25}$	$0.75_{\pm 0.13}$	$11.02_{\pm 0.77}$	$14.74_{\pm 0.51}$
TASD $_{w/\ \text{TRLoss in 2nd pass}}$	$13.89_{\pm 0.58}$	$4.78_{\pm 0.61}$	$1.47_{\pm 0.14}$	$0.52_{\pm 0.20}$	$11.07_{\pm 0.66}$	$14.73_{\pm 0.79}$
TASD	$14.19_{\pm 1.08}$	$5.17_{\pm 0.38}$	$1.71_{\pm 0.32}$	$0.78_{\pm 0.21}$	$11.65_{\pm 0.71}$	$14.96_{\pm 1.10}$

According to the results reported on the numericNLG dataset, the TRLoss is helpful in enhancing the capability of table comprehension though, the best performance is achieved by TASD $_{w/o\text{D}\ w/\ \text{TRLoss}}$. It seems that the performance improvement gained by the table comprehension enhancement is sacrificed after the text deliberation is adopted. Meanwhile, on the Totto dataset, TASD with the table reconstruction (i.e., TASD $_{w/\ \text{TRLoss}}$) does achieve the best performance in terms of BLEU-1, BLEU-2, METEOR, and ROUGE-L, though the improvement is not remarkable. The contents of the input tables are mainly linguistic and the table structures are not too diverse might be able to explain the performance improvement of TASD $_{w/\ \text{TRLoss}}$ on the Totto dataset. With the above comparisons, we can conclude that, for the input tables with diverse structures, the limitation of the current text deliberation mechanism cannot be neglected if one aims to enhance the capability of table comprehension for the table-to-text task. Moreover, this also suggests that the generalization capability of text deliberation of TASD should be improved in the future.

Limitations. In this work, long tables in the Totto dataset are removed since the efficiency and performance of TASD on large tables could be lowered. In the future, the capability of handling long tables for table-to-text models should be further explored. Besides, a large-scale and more exhaustive human evaluation is necessary. We plan to recruit more volunteers to conduct the human annotation.

7 Conclusion

In this paper, to realize table-to-text with the pretrained language model, we proposed a table structure understanding and text deliberating approach, namely TASD. The table structure understanding was realized by developing a hierarchical multi-head attention network, which can benefit the fine-tuning of the text-to-text pretrained model. The fully represented table information benefits not only the pretrained language model but also the text deliberation process since the structure information with rich semantics could be fed into the second-pass decoding naturally. We carried out extensive experiments on two public datasets with different table types. Automatic and human-based evaluations, as well as qualitative analysis, validate the effectiveness of our approach to generating faithful and fluent table descriptions. In the future, we will improve text deliberation by devising a unified framework to integrate the multi-pass decoder and refine the descriptive text paying more attention to sentence fluency.

Acknowledgements

This work is supported in part by Foshan HKUST Projects (FSUST21-FYTRI01A, FSUST 21-FYTRI02A).

References

Yang Bai, Ziran Li, Ning Ding, Ying Shen, and Hai-Tao Zheng. 2021. Infobox-to-text generation with tree-like planning based attention network. In *IJCAI*, pages 3773–3779.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *9th workshop on statistical machine translation*, pages 376–380.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186.

Rui Dong and David A Smith. 2021. Structural encoding and pre-training matter: Adapting bert for table-based fact verification. In *EACL*, pages 2366–2375.

Xinwei Geng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2018. Adaptive multi-pass decoder for neural machine translation. In *EMNLP*, pages 523–532.

Heng Gong, Xiaocheng Feng, Bing Qin, and Ting Liu. 2019a. Table-to-text generation with effective hierarchical encoder on three dimensions (row, column and time). In *EMNLP-IJCNLP*, pages 3143–3152.

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

Li Gong, Josep M Crego, and Jean Senellart. 2019b. Enhanced transformer model for data-to-text generation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 148–156.

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

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

Liang Li, Can Ma, Yinliang Yue, and Dayong Hu. 2021a. Improving encoder by auxiliary supervision tasks for table-to-text generation. In *ACL-IJCNLP*, pages 5979–5989.

Xiao Li, Yawei Sun, and Gong Cheng. 2021b. Tsqa: Tabular scenario based question answering. In *AAAI*, volume 35, pages 13297–13305.

Yangming Li and Kaisheng Yao. 2021. Rewriter-evaluator architecture for neural machine translation. In *ACL-IJCNLP*, pages 5701–5710.

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

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018. Table-to-text generation by structure-aware seq2seq learning. In *AAAI*.

Hongyuan Mei, Mohit Bansal, and Matthew R Walter. 2016. What to talk about and how? selective generation using lstms with coarse-to-fine alignment. In *NAACL-HLT*, pages 720–730.

Soichiro Murakami, Akihiko Watanabe, Akira Miyazawa, Keiichi Goshima, Toshihiko Yanase, Hiroya Takamura, and Yusuke Miyao. 2017. Learning to generate market comments from stock prices. In *ACL*, pages 1374–1384.

Jan Niehues, Eunah Cho, Thanh-Le Ha, and Alex Waibel. 2016. Pre-translation for neural machine translation. In *COLING*, pages 1828–1836.

Jason Obeid and Enamul Hoque. 2020. Chart-to-text: Generating natural language descriptions for charts by adapting the transformer model. In *INLG*, pages 138–147.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 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 *EMNLP*, pages 1173–1186.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. In *AAAI*, volume 33, pages 6908–6915.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *ACL*, pages 1073–1083.

Lya Hulliyyatus Suadaa, Hidetaka Kamigaito, Kotaro Funakoshi, Manabu Okumura, and Hiroya Takamura. 2021. Towards table-to-text generation with numerical reasoning. In *ACL-IJCNLP*, pages 1451–1465.

Ran Tian, Shashi Narayan, Thibault Sellam, and Ankur P Parikh. 2019. Sticking to the facts: Confident decoding for faithful data-to-text generation. *arXiv*.

Bayu Trisedya, Jianzhong Qi, and Rui Zhang. 2020. Sentence generation for entity description with content-plan attention. In *AAAI*, volume 34, pages 9057–9064.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*, pages 5998–6008.

Peng Wang, Junyang Lin, An Yang, Chang Zhou, Yichang Zhang, Jingren Zhou, and Hongxia Yang. 2021. Sketch and refine: Towards faithful and informative table-to-text generation. In *ACL-IJCNLP*, pages 4831–4843.

Zhenyi Wang, Xiaoyang Wang, Bang An, Dong Yu, and Changyou Chen. 2020. Towards faithful neural table-to-text generation with content-matching constraints. In *ACL*, pages 1072–1086.

Sam Wiseman, Stuart M Shieber, and Alexander M Rush. 2017. Challenges in data-to-document generation. In *EMNLP*, pages 2253–2263.

Yingce Xia, Fei Tian, Lijun Wu, Jianxin Lin, Tao Qin, Nenghai Yu, and Tie-Yan Liu. 2017. Deliberation networks: Sequence generation beyond one-pass decoding. *NeurIPS*, 30:1784–1794.

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

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. Tabert: Pretraining for joint understanding of textual and tabular data. In *ACL*, pages 8413–8426.

Hongzhi Zhang, Yingyao Wang, Sirui Wang, Xuezhi Cao, Fuzheng Zhang, and Zhongyuan Wang. 2020. Table fact verification with structure-aware transformer. In *EMNLP*, pages 1624–1629.

A Human Evaluation Settings

The criteria adopted in our human-based evaluation are (1) **Grammar** (e.g., is this paragraph grammatical?), (2) **Coherence & Concise** (e.g., is this paragraph coherent and contextually consistent? does it repeat redundant information?), and (3) **Factual perspective** (e.g., are the facts that this paragraph describes correct? are these facts related to references and tables?). More specifically, we list the detailed justifications on how to score the generated text in each criterion as follows.

Grammar

- **1** It is more like garbled code than a paragraph.
- **2** There are many obvious grammatical mistakes.
- **3** There are a few obvious grammatical mistakes.
- **4** There are few grammatical mistakes.
- **5** There are no grammatical mistakes.

Coherence & Concise

- **1** The logic of text expression is chaotic and nonsense.
- **2** There are a lot of logical inconsistencies or redundant information.
- **3** There are some logical inconsistencies or redundant information.
- **4** There are a few logical inconsistencies or redundant information, but it does not affect browsing.
- **5** The logic of the text is smooth without redundant information.

Factual Perspective

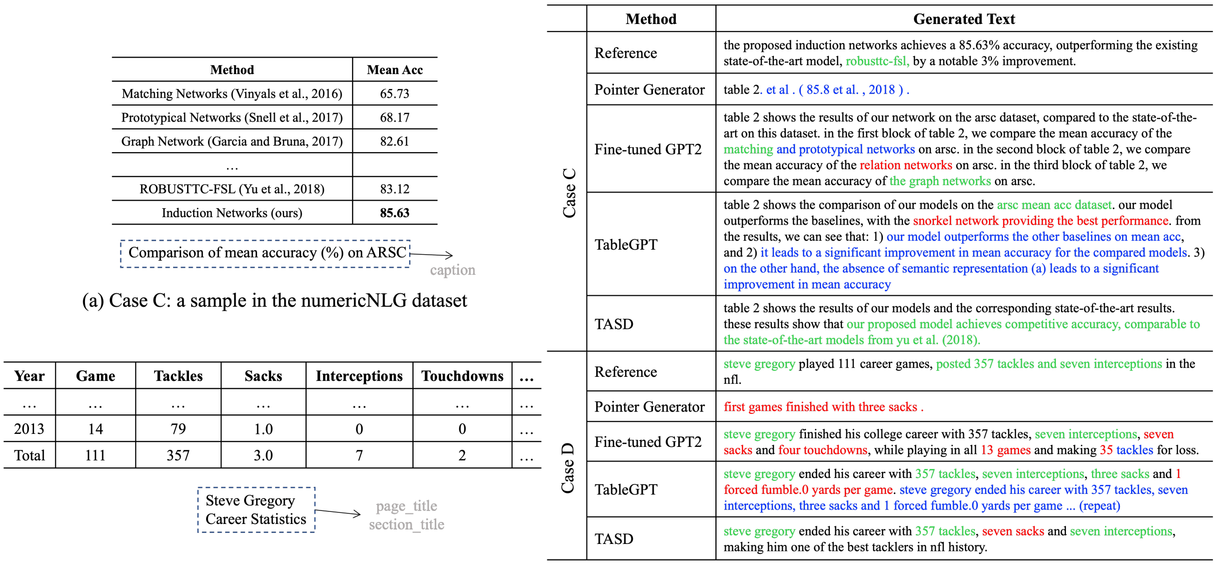
- **1** The paragraph does not coincide with the reference or table, and it is full of information inconsistent with the facts.
- **2** The paragraph describes the facts incorrectly and has a low correlation with reference, but is related to the information in the table.
- **3** The paragraph description is incorrect, but it is highly coincident with the reference.
- **4** The paragraph description is basically correct, and the coincidence with the reference is low, but it also describes the information in the table.
- **5** The paragraph description is correct and highly coincident with the reference.

B Illustrative Examples of Generated Descriptions

We additionally selected another two examples of the generated table descriptions from the numericNLG and Totto datasets, respectively. The results are shown in Figs. 6 and 7. From these four examples, we can see that TASD can generate more accurate and fluent descriptive texts. While incorrect descriptions can be found in the outcome texts generated by different models for cases D and F, which suggests that generating faithful descriptions for open-domain tables is much more challenging and requires more powerful and, thus larger, pretrained language models.

C Extra Implementation Details

The learning rate of GPT2 was searched from $\{3e-4, 3e-5, 3e-6\}$. In the evaluation of discussing the limitation of text deliberation (see Section 6), a trade-off parameter for balancing the GPT2 finetuning loss and the TRLoss was adopted, then the trade-off parameter was searched from $\{1e-1, 5e-2, 1e-2, 5e-3, 1e-3\}$, and 1e-2 was selected for the reported performance. Besides, the reported results in Tables 4 and 5 were averaged in 3 runs.



Method	Mean Acc
Matching Networks (Vinyals et al., 2016)	65.73
Prototypical Networks (Snell et al., 2017)	68.17
Graph Network (Garcia and Bruna, 2017)	82.61
…	
ROBUSTTC-FSL (Yu et al., 2018)	83.12
Induction Networks (ours)	**85.63**

Comparison of mean accuracy (%) on ARSC

(a) Case C: a sample in the numericNLG dataset

Year	Game	Tackles	Sacks	Interceptions	Touchdowns	…
…	…	…	…	…	…	…
2013	14	79	1.0	0	0	…
Total	111	357	3.0	7	2	…

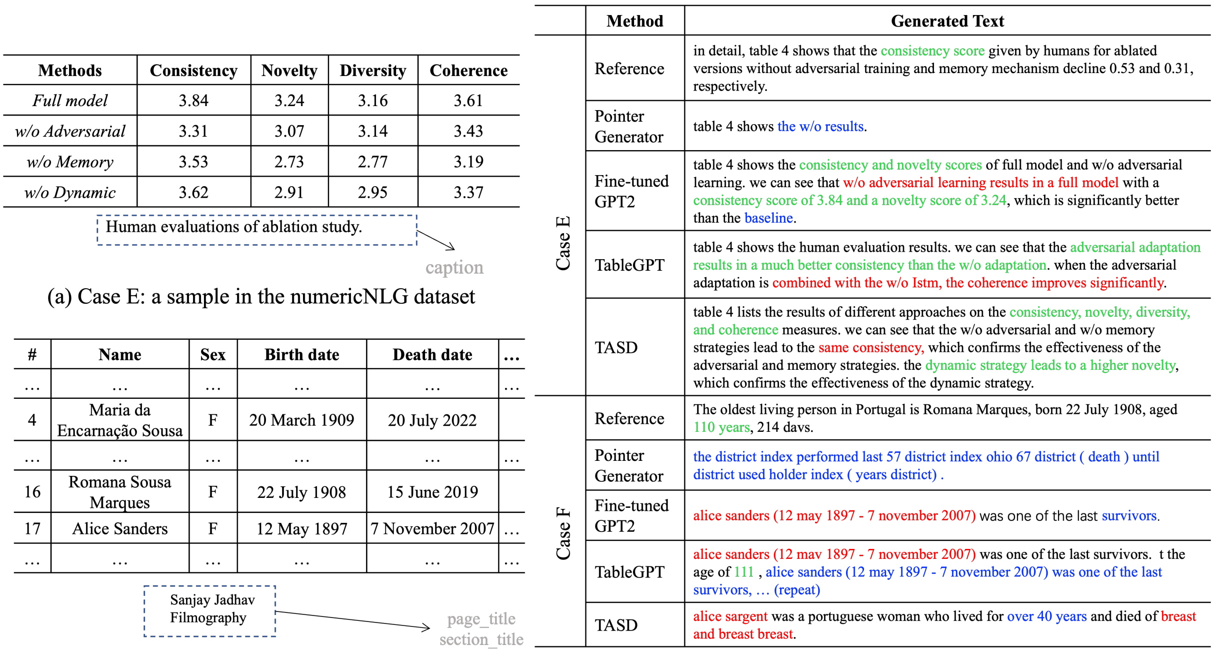
Steve Gregory
Career Statistics

(b) Case D: a sample in the Totto dataset

	Method	Generated Text
Case C	Reference	the proposed induction networks achieves a 85.63% accuracy, outperforming the existing state-of-the-art model, robusttc-fsl, by a notable 3% improvement.
	Pointer Generator	table 2. et al . (85.8 et al. , 2018) .
	Fine-tuned GPT2	table 2 shows the results of our network on the arsc dataset, compared to the state-of-the-art on this dataset. in the first block of table 2, we compare the mean accuracy of the matching and prototypical networks on arsc. in the second block of table 2, we compare the mean accuracy of the relation networks on arsc. in the third block of table 2, we compare the mean accuracy of the graph networks on arsc.
	TableGPT	table 2 shows the comparison of our models on the arsc mean acc dataset. our model outperforms the baselines, with the snorkel network providing the best performance. from the results, we can see that: 1) our model outperforms the other baselines on mean acc, and 2) it leads to a significant improvement in mean accuracy for the compared models. 3) on the other hand, the absence of semantic representation (a) leads to a significant improvement in mean accuracy
	TASD	table 2 shows the results of our models and the corresponding state-of-the-art results. these results show that our proposed model achieves competitive accuracy, comparable to the state-of-the-art models from yu et al. (2018).
Case D	Reference	steve gregory played 111 career games, posted 357 tackles and seven interceptions in the nfl.
	Pointer Generator	first games finished with three sacks .
	Fine-tuned GPT2	steve gregory finished his college career with 357 tackles, seven interceptions, seven sacks and four touchdowns, while playing in all 13 games and making 35 tackles for loss.
	TableGPT	steve gregory ended his career with 357 tackles, seven interceptions, three sacks and 1 forced fumble.0 yards per game. steve gregory ended his career with 357 tackles, seven interceptions, three sacks and 1 forced fumble.0 yards per game ... (repeat)
	TASD	steve gregory ended his career with 357 tackles, seven sacks and seven interceptions, making him one of the best tacklers in nfl history.

(c) Generated texts for case C and case D

Figure 6: Generated table descriptions on cases C and D.



Methods	Consistency	Novelty	Diversity	Coherence
Full model	3.84	3.24	3.16	3.61
w/o Adversarial	3.31	3.07	3.14	3.43
w/o Memory	3.53	2.73	2.77	3.19
w/o Dynamic	3.62	2.91	2.95	3.37

Human evaluations of ablation study.

(a) Case E: a sample in the numericNLG dataset

#	Name	Sex	Birth date	Death date	…
…	…	…	…	…	…
4	Maria da Encarnação Sousa	F	20 March 1909	20 July 2022	
…	…	…	…	…	…
16	Romana Sousa Marques	F	22 July 1908	15 June 2019	
17	Alice Sanders	F	12 May 1897	7 November 2007	…
…	…	…	…	…	…

Sanjay Jadhav
Filmography

(b) Case F: a sample in the Totto dataset

	Method	Generated Text
Case E	Reference	in detail, table 4 shows that the consistency score given by humans for ablated versions without adversarial training and memory mechanism decline 0.53 and 0.31, respectively.
	Pointer Generator	table 4 shows the w/o results.
	Fine-tuned GPT2	table 4 shows the consistency and novelty scores of full model and w/o adversarial learning. we can see that w/o adversarial learning results in a full model with a consistency score of 3.84 and a novelty score of 3.24, which is significantly better than the baseline.
	TableGPT	table 4 shows the human evaluation results. we can see that the adversarial adaptation results in a much better consistency than the w/o adaptation. when the adversarial adaptation is combined with the w/o lstm, the coherence improves significantly.
	TASD	table 4 lists the results of different approaches on the consistency, novelty, diversity, and coherence measures. we can see that the w/o adversarial and w/o memory strategies lead to the same consistency, which confirms the effectiveness of the adversarial and memory strategies. the dynamic strategy leads to a higher novelty, which confirms the effectiveness of the dynamic strategy.
Case F	Reference	The oldest living person in Portugal is Romana Marques, born 22 July 1908, aged 110 years, 214 davs.
	Pointer Generator	the district index performed last 57 district index ohio 67 district (death) until district used holder index (years district) .
	Fine-tuned GPT2	alice sanders (12 may 1897 - 7 november 2007) was one of the last survivors.
	TableGPT	alice sanders (12 mav 1897 - 7 november 2007) was one of the last survivors. t the age of 111 , alice sanders (12 may 1897 - 7 november 2007) was one of the last survivors, … (repeat)
	TASD	alice sargent was a portuguese woman who lived for over 40 years and died of breast and breast breast.

(c) Generated texts for case E and case F

Figure 7: Generated table descriptions on cases E and F.