TableGPT: Few-shot Table-to-Text Generation
with Table Structure Reconstruction and Content Matching

Heng Gong[1*], **Yawei Sun**[1*], **Xiaocheng Feng**[1,2], **Bing Qin**[1,2],
Wei Bi, **Xiaojiang Liu**, **Ting Liu**[1,2]

[1]Department of Computer Science and Technology,
Harbin Institute of Technology, Harbin, China
[2]Peng Cheng Laboratory

{hgong,ywsun,xcfeng,qinb,tliu}@ir.hit.edu.cn
weibi@cse.ust.hk   xiaojiangliu84@hotmail.com

Abstract

Although neural table-to-text models have achieved remarkable progress with the help of large-scale datasets, they suffer insufficient learning problem with limited training data. Recently, pre-trained language models show potential in few-shot learning with linguistic knowledge learnt from pretraining on large-scale corpus. However, benefiting table-to-text generation in few-shot setting with the powerful pretrained language model faces three challenges, including (1) the gap between the task's structured input and the natural language input for pretraining language model. (2) The lack of modeling for table structure and (3) improving text *fidelity* with less incorrect expressions that are contradicting to the table. To address aforementioned problems, we propose TableGPT for table-to-text generation. At first, we utilize table transformation module with template to rewrite structured table in natural language as input for GPT-2. In addition, we exploit multi-task learning with two auxiliary tasks that preserve table's structural information by reconstructing the structure from GPT-2's representation and improving the text's fidelity with content matching task aligning the table and information in the generated text. By experimenting on *Humans*, *Songs* and *Books*, three few-shot table-to-text datasets in different domains, our model outperforms existing systems on most few-shot settings.

1 Introduction

Table-to-text generation, aiming at generating descriptive text about important information in structured data, has well application prospect in communicating with human in a comprehensible and natural way, such as financial report (Murakami et al., 2017), medical report (Hasan and Farri, 2019) generation, etc. In recent years, data-driven models have shown impressive capability to produce informative and fluent text with the help of large-scale datasets, such as WIKIBIO (Lebret et al., 2016) and E2E (Dušek et al., 2020). However, it is not always feasible to collect large-scale labeled dataset for various domains in the real world, resulting in unsatisfying performance due to the insufficient training. Such few-shot learning setting for table-to-text generation is not well-explored, and in this paper, we focus on exploring how to efficiently model for few-shot table-to-text generation with limited training pairs.

Recently, pre-trained language models have shown promising progress in various natural language processing tasks (Yang et al., 2019b; Devlin et al., 2019; Radford et al., 2019). They can capture linguistic knowledge by pretraining on large-scale unlabeled dataset and generalize to downstream tasks with little labeled data in target domain, effectively modeling for few-shot setting (Peng et al., 2020). However, efforts to benefit table-to-text generation from the powerful pre-trained language model, especially in few-shot setting, are non-trivial due to three challenges. (1) There is a gap between the structured data input for table-to-text generation and natural language input that is used for pretraining GPT-2. (2) Also, it lacks modeling of the table's structure which contains rich information to understand the input before generating text. (3) Additionally, it doesn't address how to maintain text's *fidelity* for table-to-text gener-

---

Attribute	Value
name	james beattie
fullname	james scott beattie
birth	27 february 1978 lancaster , england
position	striker
currentclub	swansea city (assistant first team coach)
nationalteam	england u21 england
article	james beattie (footballer)
...	...

Reference: james scott beattie (born 27 february 1978) is an english former professional footballer who played as a striker .

Table 1: A sample of table-text pair for table-to-text generation from *Humans* domain. The highlighted phrases are *information* mentioned in text.

ation while exploiting linguistic knowledge from pretraining corpus, that is the (highlighted) information in text (Table 1) should correctly derive from structured data.

In order to alleviate aforementioned problems, we propose TableGPT that focus on generating high-fidelity text for table-to-text generation with limited training pairs. Addressing the gap between structured table input and natural language input that GPT-2 processes during pretraining, we utilize a table transformation module that employs template to naturally transform structured table into natural language. In addition, we utilize two auxiliary tasks under the framework of multi-task learning, table structure construction and content matching, targeting pretrained GPT-2's lack of modeling for table structure and text's fidelity. In detail, the table structure reconstruction task is proposed for GPT-2 which force it to embed table structure into its representation when modeling structured table. Besides, we utilize content matching task that help model correctly describe important information from table via Optimal-Transport (Chen et al., 2019a) technique, which measures the distance between the information in generated text and information in table and use the distance as penalty for text with incorrect information.

We conducted experiments on three data-to-text datasets on different domains (Chen et al., 2020b): *Humans*, *Books*, *Songs* in various settings. Both automatic evaluation and human evaluation results show that our model can achieve new state-of-the-art performance for table-to-text generation in terms of generating fluent and high-fidelity text in most few-shot settings.

2   Background

2.1   Task Definition

For the table-to-text task discussed in this paper, we can formulate each training instance as pair of table and summary $E = (S, T)$. Given a table, which can be formulated as sets of records $S = \{r_i\}_{i=1}^N$, the model is expected to generate descriptive text $T = w_1, w_2, ..., w_L$. N is the number of records and L is the number of words in text. Each record r_i consists of two type of information: $r_i.a$ and $r_i.v$. $r_i.a$ denotes the attribute of the record (e.g. name) and $r_i.v$ denotes the corresponding value (e.g. james beattie). Please note that both $r_i.a$ and $r_i.v$ can be viewed as a sequence of words.

2.2   Pre-trained Language Model

Recently, pre-trained language models, such as BERT (Devlin et al., 2019), GPT-2 (Radford et al., 2019), XLNet (Yang et al., 2019b) and more, have achieved remarkable progress in various NLP tasks. The main idea is to pretrain a neural language model with large number of parameters on large-scale dataset in order to capture the linguistic knowledge at first. Then, transfer those knowledge to downstream task
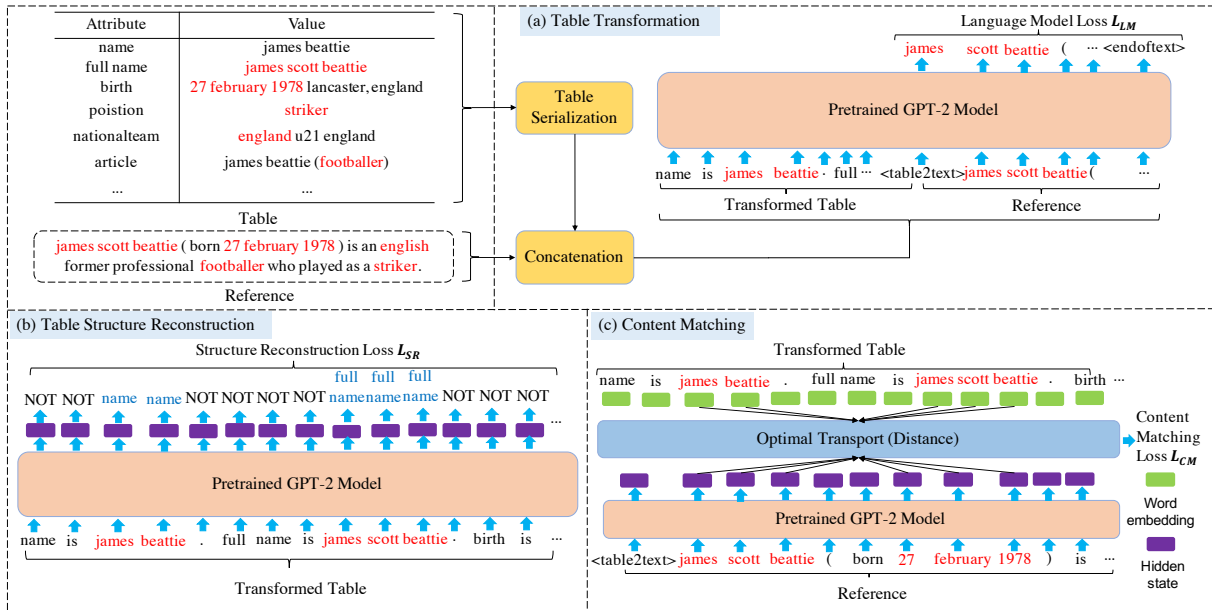
Figure 1: TableGPT's Training Process. Tokens in red indicate information in reference and those in blue refers to value tokens' corresponding attribute. (a) Table Transformation employs template to rewrite the structured table in natural language and concatenate with reference to form a training pair, resulting in language model loss L_{LM} with GPT-2. (b) Table Structure Reconstruction and (c) Content Matching are auxiliary tasks. The former reconstructs attributes from GPT-2's representation of value tokens, forming L_{SR}. The latter use Optimal Transport to measure distance between information mentioned in text and table, using this as content matching loss L_{CM}. The model is jointly finetuned with above three losses.

via finetuning on the task's dataset. Impressively, they can outperform various NLP tasks' previous state-of-the-arts by a large margin. Since we investigate table-to-text generation task in this paper instead of natural language understanding tasks, we choose GPT-2 as the basis of our model. The model structure for GPT-2 is a 12-to-48-layer transformer decoder (Vaswani et al., 2017) with 117 million to 1542 million parameters. Each layer consists of a stack of masked multi-head self attention and feed-forward neural network with residual connection and layer normalization. The pre-training target of GPT-2 is as the same as language model: maximizing gold text's probability. The large-scale model is trained on vast and diverse WebText dataset with 8 million documents collected from the Internet. Its success in the area of text generation attributes to both its high capacity model and knowledge learnt from pretraining on vast dataset.

3 Approach

In this section, we propose to address the three incompatibility between pretrained language model and table-to-text generation illustrated in Section 1. Figure 1 presents the overall multi-task training framework of our model. We first utilize table transformation module to reasonably transform structured table into text sequence and aggregate it with reference text, resulting in a suitable training sequence for GPT-2 model. Then, two auxiliary tasks: table structure reconstruction and content matching with optimal transport (Chen et al., 2019a) are performed on top of GPT-2's representation of the training sequence. Those two auxiliary tasks' training objectives along with GPT-2's language model training objective are jointly finetuned based on the pre-trained GPT-2 under the multi-task training framework. The overall objective here is to produce high-fidelity text while maintaining its fluency.

3.1 Table Transformation

As noted in Section 2.1, the given structured table consists of multiple records as attribute-value pairs. In order to adapt to the sequential nature of language model, we employ a template-based table serialization

method (Chen et al., 2019b) to encode such structured table as a sequence. For example, we serialized the attribute-value pair "name: jack reynolds" as a sentence "name is jack reynolds." and concatenate all of them into a document according to the order of records in table.

After obtaining the serialized structured table, we connect it with the corresponding natural language description T with special token "$< table2text >$". It serves as a functional token that both encodes the overall information of the table and as a starting signal to generate text. The whole sequence is ended with special token "$< endoftext >$". In this way, our model encodes the structured table and learns to predict the target sequence one word at a time as in GPT-2. We denote the final input sequence as $ST = st_1, \cdots, st_{m+n+2}$. m is the length of serialized table, n is the length of text and 2 refers to the special tokens mentioned above. The language model's training objective is to maximize the likelihood of the reference text, which is equivalent to minimize the negative log likelihood (language model loss $\mathcal{L}_{LM}$) as characterized by Equation 1.

$$\mathcal{L}_{LM} = -\frac{1}{n+1} \sum_{i=m+2}^{m+n+2} logP(st_i|st_{<i}) \tag{1}$$

3.2 Table Structure Reconstruction

As shown in Table 1, unlike many natural language generation tasks that take sentences as input, table-to-text generation models need to process table with structural information. Each data record in the input can be seen as a pair of attribute and value. Traditionally, table-to-text models utilized attribute-value concatenation to represent the tables. In this way, they are able to capture the structural information by learning the correspondence between value and attribute. However, when we transform the table into natural language and use pre-trained language model GPT-2 for representation, it lacks the explicit modeling to incorporate such structural information. Inspired by Liu et al. (2019), our model treats the attribute names as the labels for the model to reconstruct such structural information from GPT-2's learned table values' representation.

In detail, as shown in Figure 1 (b), given a serialized table S_k which consists of different attributes and values in natural language form, the table structure reconstruction task takes the last layer of GPT-2's hidden states for each value tokens $[H^{t_{i,j}}]_{i=1:n,j=1:m_i}$ of the table and classify which attribute does each value token's representation refer to. Specifically, the i means the ith record of the table S_k, j means the jth value token of ith record, n is the number of records and m_i is the number of tokens of ith record's value.

$$P(a_{i,j}) = \mathbf{softmax}(\mathbf{H^{t_{i,j}}W_t + b_t}) \tag{2}$$

$$\mathcal{L}_{SR} = -\frac{1}{Z} \sum_{i=1}^{n} \sum_{j=1}^{m_i} logP(a_{i,j}^{\star}) \tag{3}$$

Equation 2 shows the detail of the reconstruction classifier. Please note that the serialized table consists of attribute, value and template tokens. In this auxiliary task, we only take the GPT-2's hidden states for those value tokens and reconstruct the structural information by classifying their corresponding attribute. W_t and b_t are the trainable parameters of the introduced classifier and $p(a_{i,j})$ is probability to classify value token $H^{t_{i,j}}$ as referring to attribute $a_{i,j}$. We use cross entropy as this task's objective function, illustrated by Equation 2 and 3. $a_{i,j}^{\star}$ refers to the gold attribute label for the value token and Z is the number of value tokens' that need to reconstruct the corresponding attribute label. By incorporating this auxiliary task, TableGPT can be guided to embed structural information when representing the table at the training stage.

3.3 Content Matching

Take Table 1 as an example, generating high fidelity text that correctly describe information in the table is the core of table-to-text generation. Producing fluent but incorrect text still means unsatisfying performance as the text is not reliable for the purpose of disseminating comprehensible information. Ideally,

when generating words that is intended to describe information in table, directly copying them from table will result in high-fidelity text. However, it's non-trivial to integrate a copy mechanism inside the transformer architecture of GPT-2 model, since the change of model structure may break syntactic and semantic features contained in the pretrained language model, which are essential for text generation especially in few-shot setting. Also, rephrasing sometimes is needed to produce more natural text.

In order to encourage our model to generate high-fidelity text while keeping GPT-2's advantage of produing fluent text, we utilize another auxiliary task, called content matching task, during finetuning on the table-to-text corpus. The content matching task is to explicitly match the important information in a table with information in the corresponding generated text. An intuitive way is to apply a mis-matching loss by hard-matching key information in table and information in the generated text. But that is discrete and non-differentiable and the corresponding gradient descent can't be learned directly. Inspired by optimal transport (OT) that can measure the distance between information in source sequence and target sequence (Chen et al., 2019a) without breaking the end-to-end training process, we adopt it as a content matching loss that guide the model to generate text containing information that align with the table.

As in Section 3.1, the whole GPT-2 training sequence consists of serialized table and reference text. The serialized table sequence, $x = x_1, \cdots, x_m$, can be represented as a discrete distribution $\mu = \sum_{i=1}^{m} u_i \delta_{x_i}$, where $u_i \geq 0$ and $\sum_i u_i = 1$, m is the length and δ_x is the Dirac function centered on x. Similar with the serialized table sequence, the discrete distribution of reference text sequence $y = y_1, \cdots, y_n$ can be represented as $\nu = \sum_{j=1}^{n} v_j \delta_{y_j}$. Under such setting, computing the OT (optimal transport) distance between probability distributions $\mathbf{u} = \{u_i\}_{i=1}^{m}$ and $\mathbf{v} = \{v_j\}_{j=1}^{n}$ is defined as the solution of the following network-flow problem (Luise et al., 2018):

$$\mathcal{L}_{CM} = \min_{\mathbf{T} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \sum_{i=1}^{m} \sum_{j=1}^{n} T_{ij} \cdot d(x_i, y_j) \tag{4}$$

where $\Pi(\boldsymbol{\mu}, \boldsymbol{\nu}) = \{\mathbf{T} \in \mathbb{R}_{+}^{m \times n} | \mathbf{T}\mathbf{1}_n = \boldsymbol{\mu}, \mathbf{T}^{\top}\mathbf{1}_m = \boldsymbol{\nu}\}$ which is the set of joint distribution of the two marginal distribution $\mathbf{u}$ and $\mathbf{v}$, $\mathbf{1}_n$ and $\mathbf{1}_m$ are n-dimensional all-one vector and m-dimensional all-one vector respectively, and $d(x_i, y_j)$ denotes the cost of moving x_i to y_j. Especially, we adopt the cosine distance between two token embedding vectors of x_i and y_j, which is defined as $d(x_i, y_j) = 1 - \frac{x_i y_j}{\|x_i\|_2 \|y_j\|_2}$. Exact minimization over $\mathbf{T}$ is computational intractable. In order to overcome this problem, we use the recently proposed Inexact Proximal point method for Optimal Transport (IPOT) (Chen et al., 2019a) as an approximation.

For natural language generation tasks such as neural machine translation, OT distance is often applied by matching source sequence with whole target sequence, since almost every word in both sequences are supposed to be matched. However, when it comes to table-to-text task in a realistic way, there are some redundant information or words in both table and text. In order to apply the OT distance, unlike previous adoption (Wang et al., 2020) based on the assumption that all information in the table should be described in text, we propose to only match the record words which appear in both table and reference text. In this way, the OT distance is able to avoid wrongly penalizing text that doesn't mention redundant information in table.

3.4 Learning Objective

For table structure reconstruction and content matching, both auxiliary tasks are trained with the main GPT-2's language model loss together, which can be regarded as multi-task learning. The loss function of multi-task learning consists of language model loss $\mathcal{L}_{LM}$, table structure reconstruction loss $\mathcal{L}_{SR}$ and content matching loss $\mathcal{L}_{CM}$. In this way, the loss function $\mathcal{L}_{MT}$ of the full model is computed as follows:

$$\mathcal{L}_{MT} = \mathcal{L}_{LM} + \lambda_1 \mathcal{L}_{SR} + \lambda_2 \mathcal{L}_{CM}, \tag{5}$$

where λ_1 and λ_2 are hyper-parameters that are two scale factors. Please note that when optimizing the $\mathcal{L}_{CM}$ with IPOT algorithm, the gradients of OT loss are hard to back propagate to model's parameters, since the process of sampling words from multinomial distribution which comes from language model

Domain	Humans				Books				Songs			
# of training instances	50	100	200	500	50	100	200	500	50	100	200	500
Base	2.9	5.1	6.1	8.3	7.3	6.8	7.8	8.8	10.4	12.0	11.6	13.1
Base + switch + LM	25.7	29.5	36.1	41.7	34.3	36.2	37.9	40.3	36.1	37.2	**39.4**	42.2
Base + switch + LM (R)	25.1	27.7	33.7	41.6	33.8	35.4	37.6	40.9	35.3	**37.8**	39.2	41.8
TableGPT	**29.8**	**34.5**	**40.6**	**45.6**	**35.1**	**37.3**	**38.5**	41.6	**36.7**	**37.8**	39.3	**42.3**
-sr	29.1	34.0	40.2	45.5	34.8	37.1	37.1	41.5	35.6	37.7	39.2	41.8
-cm	29.2	33.5	40.0	**45.6**	34.8	37.0	36.5	**42.2**	35.7	37.4	39.2	41.8
-sr&cm	29.0	32.9	40.0	45.3	34.2	36.5	38.4	41.4	35.5	37.0	39.0	41.6

Table 2: BLEU-4 results on *Humans*, *Books* and *Songs* domains.

Domain	Humans				Books				Songs			
# of training instances	50	100	200	500	50	100	200	500	50	100	200	500
Base	0.1	0.4	0.8	1.5	1.7	1.5	2.1	2.4	4.1	5.1	4.7	5.8
Base + switch + LM	14.1	16.2	22.1	28.3	22.5	23.1	25.0	27.6	26.2	28.6	30.1	32.6
Base + switch + LM (R)	12.1	14.1	19.1	28.2	22.2	22.2	24.4	27.3	27.0	28.3	28.8	32.0
TableGPT	16.3	**20.6**	**27.6**	**32.4**	**24.0**	25.4	**26.7**	28.9	**27.1**	**29.4**	30.6	32.8
-sr	**16.7**	20.2	26.2	32.3	23.4	**25.7**	26.5	29.2	26.3	29.0	**31.0**	33.5
-cm	15.7	20.0	25.4	31.6	22.3	24.9	26.3	**29.7**	26.2	28.6	30.9	33.1
-sr&cm	16.2	19.3	26.6	32.1	23.7	25.1	26.4	28.7	25.8	25.9	30.6	**33.7**

Table 3: ROUGE-4 (F-measure) results on *Humans*, *Books* and *Songs* domains.

is unfortunately non-differentiable. In order to back propagate $\mathcal{L}_{CM}$, we adopt the Soft-argmax trick to approximate each word embedding vector. During testing, we feed the serialized table with special token $< table2text >$ into GPT-2 and generate the text word by word.

4 Experiment

4.1 Datasets and Experiment Settings

Following Chen et al. (2020b)'s work on few-shot table-to-text generation, we conducted experiments on three datasets in different domains: *Humans*, *Books* and *Songs* respectively. Each instance in those datasets consists of the structured infobox as the table and first sentence of the article as reference. Same as Chen et al. (2020b), we train our model on different few-shot setting with training dataset size varying from 50, 100, 200 to 500. Also, models are chosen based on performance on validation set with 1000 instances. Test sets for *Humans*, *Books* and *Songs* consist of 13587, 5252 and 11879 instances respectively.

We implement TableGPT based on the transformers library (Wolf et al., 2019). The configuration of base GPT-2 model is 12 layers and 8 attention heads per layer. For optimizer, we adopt the OpenA-IAdamW optimizer with 100 warm steps. We train the model with learning rate set to 2e-4. The batch size is set to 10 for all datasets. The weights λ_1, λ_2 of the table structure reconstruction loss and content matching loss are both 0.2 according to performance on validation set. Following Chen et al. (2020b)'s way to deal with the vocabulary limitation, for all datasets we use the Byte Pair Encoding (BPE) and subword vocabulary as in Radford et al. (2019).

4.2 Comparing Methods

We compare our proposed **TableGPT** with baseline model and previous state-of-the-art model: **Base** and **Base+switch+LM**. More details can be found in Chen et al. (2020b).

Domain	Humans			Books			Songs		
Metrics	#Sup	#Cont	Nat.	#Sup	#Cont	Nat.	#Sup	#Cont	Nat.
Reference	4.20*	0.36*	1.27*	3.54*	0.43	1.03*	3.22	0.64*	1.26*
Base+switch+LM(R)	3.20*	0.96*	0.85	3.32	0.46	1.15*	2.92	0.75	0.73*
TableGPT	3.67	0.60	0.89	3.28	0.42	0.83	3.05	0.77	1.01

Table 4: Human evaluation results. Models with * perform significantly different from TableGPT ($p <$ 0.05), using a one-way ANOVA with posthoc Tukey HSD tests.

- **Base:** It is based on a Seq2Seq model with field-gating encoder that incorporate the table structure's information during encoding (Liu et al., 2018). Additionally, it utilizes the pre-trained word embedding which is fixed during the training stage. Since it achieves competitive performance on large-scale dataset, it can show how a data-driven Seq2Seq model typically perform with limited training data.
- **Base + switch + LM:** It tries to exploit GPT-2's learnt knowledge from pretraining on vast corpus by proposing a switch policy that choosing between copying from infobox and generating from the GPT-2 language model when generating each word in text. We also use the released codes and data[1] by Chen et al. (2020b) to reproduce its result for human evaluation and report corresponding automatic evaulation results as **Base + switch + LM(R)**.
- **TableGPT:** In this paper, we propose TableGPT that exploits GPT-2's learnt knowledge from pretraining on vast corpus for few-shot learning while enhance it for generating high fidelity text with two auxiliary tasks. Also, we perform ablation studies for evaluating each auxiliary task's contribution. *-sr* represents the variant without table structure reconstruction, *-cm* represents the variant without content matching and *-sr&cm* shows the performance of GPT-2 without auxilary tasks.

4.3 Automatic Evaluation

Following the previous work Chen et al. (2020b), we adopt BLEU-4 (Papineni et al., 2002) and ROUGE-4 (Lin, 2004) to conduct automatic evaluations. Table 2 and 3 show corresponding results of comparing methods on different datasets. Although the **Base** achieves competitive results when training on large-scale dataset (Liu et al., 2018), the performance drops drastically in few-shot setting. While utilizing a switch policy to combine copying words from table and generating words from GPT-2 (**Base + switch + LM**) can achieve impressive performance in all few-shot setting, a standard GPT-2 model (**TableGPT - sr&cm**) that takes a serialized table as input and generate text afterwards without copying can actually perform better in most of the few-shot setting. **TableGPT** with table structure reconstruction and content matching that preserves structural information during encoding and guide the model to generate high-fidelity text can further improve the performance. Ablation studies also show that each of the auxiliary task attributes to the performance enhancement and applying both of them can achieve the best performance in most of the few-shot setting.

4.4 Human Evaluation

Following the settings in Chen et al. (2020b), we conduct human evaluation on **TableGPT** with previous state-of-the-art model **Base+switch+LM(R)** and **Reference** from two aspects: *Factual correctness* and *Language naturalness*. We sampled 100 tables along with corresponding generated text from *Humans*, *Books* and *Songs* test set (under the few-shot setting of 100 training data) respectively, resulting in 300 tables in total. In order to reduce variance caused by human, each example is evaluated by three different graduates who have passed intermediate English test and the scores are averaged in Table 4.

The first evaluation criteria *Factual correctness* focuses on how well the text can correctly describe information provided by the corresponding table. Each rater was provided with both the text and table. They are asked to count how many facts contained in the text are consistent with information from table,

---

[1]https://github.com/czyssrs/Few-Shot-NLG

Attribute	Value
name	`james beattie`
fullname	`james scott beattie`
birth	`27 february 1978` `lancaster` , `england`
height	6 1
position	`striker`
currentclub	swansea city (assistant first team coach)
youthyears	1995 – 1996
youthclubs	blackburn rovers
totalcaps	443
totalgoals	131
nationalyears	1996 2003
nationalteam	`england` u21 england
nationalcaps	9 5
nationalgoals	4 0
manageryears	2013 – 2014
managerclubs	accrington stanley
article	james beattie (`footballer`)
...	...

Reference: `james scott beattie` (born `27 february 1978`) is an `english` former professional `footballer` who played as a `striker` .

Base+switch+LM(R): `james beattie` (, born 10 july 1971) is an `english` former professional association `footballer` who played for , among others , England .

TableGPT: `james beattie` (born `27 february 1978` in `lancaster`) is a former `english` `footballer` who played as a `striker` .

Table 5: A sample table from *Human* test set and the texts generated with different methods, trained with 100 training data. The facts that both exist in source table and summary text are highlighted with the same color in background. The red text indicates the wrong fact of generated text conflicts with information in source table. The blue text indicates the generated text fragment has grammatical error.

noted as #Sup , and how many are contradicting with or missing from the table, noted as #Cont. We report the average number of supporting facts (#Sup) and contradicting facts (#Cont) on different dataset in Table 4.

The second evaluation criteria *Language naturalness* tries to evaluate these models from the aspect of grammaticality (is the sentence grammatically correct?) and fluency of the text (is the sentence fluent and natural?). We arrange text from different models on the same table into 3 pairs. For each pair of text without table, raters are asked to decide which one is better or whether both text are of same quality, solely in terms of language naturalness. When a generated summary is chosen as the better one, we assign 1.0 score to the better one and 0.0 score to the worse. If two summaries are deemed of same quality, we assign 0.5 score to both of them. We then calculate the average scores and report results on different dataset in Table 4.

Results show that **TableGPT** can produce less contradicting facts than previous state-of-the-art model **Base+switch+LM(R)** on *Humans* and *Books* and achieves comparable performance on *Songs*. Meanwhile, **TableGPT** can include more supporting facts on *Humans* and *Songs* and generate more natural text than **Base+switch+LM(R)** on *Humans* and *Songs*. Overall, our **TableGPT** model can improve text fidelity while preserving the naturalness of the text.

4.5 Case Study

Compared with the previous state-of-the-art model **Base+switch+LM(R)** and **Reference**, **TableGPT** performs better. It can accurately describe most of the key information in fluent text compared with reference. For **Base+switch+LM(R)**, the design of separate copy mechanism and GPT-2 may attributes to the inconsistent expression "played for , among others , England ." and the expression of wrong birth date, which shows the imperfect switch policy on deciding when to copy from table can sometimes hurt model's ability to generate high-fidelity text. On the contrary, TableGPT, enhanced to generate high-fidelity text with two auxiliary tasks without breaking one unified GPT-2 model for generating text, performs better in terms of fidelity and fluency of text in this example.

5 Related Work

In recent years, neural models for generating texts directly from preprocessed data (Wiseman et al., 2017; Puduppully et al., 2019a; Puduppully et al., 2019b; Gong et al., 2019; Feng et al., 2020), have become mainstream for table-to-text generation and achieved impressive performance with the help of large-scale dataset. Mei et al. (2016) proposes a pre-selector on encoder-aligner-decoder model for generation, which strengthens model's content selection ability and obtains considerable improvement over standard Seq2Seq model. Sha et al. (2018) proposes a hybrid attention mechanism for modeling the order of content when generating texts. Liu et al. (2018) presents a field-gating encoder focusing on modeling table structure and dual attention mechanism to utilize the structure information when decoding. In addition, Bao et al. (2018) develops a table-aware sequence-to-sequence model on this task. However, Chen et al. (2020b) shows that the well-performed Seq2Seq model trained on large-scale dataset suffer from limited training data in few-shot setting.

Recently, GPT-2 has been successfully adapted to dialogue generation in few-shot setting (Zhang et al., 2020; Peng et al., 2020), showing potential to address insufficient training data problem for few-shot learning with the help of learnt knowledge from pretraining on vast corpus. As for table-to-text generation, Chen et al. (2020b) propose a switch model that use GPT-2 to generate template-like functional words while generating factual expressions via copying records' values from table in few-shot scenario. Different from this work, we model the table and generate text within a GPT-2 model in a unified way and we show that our proposed TableGPT can perform well in the few-shot scenario. In addition, different from both works mentioned above, we enhance GPT-2's ability to model table structure and to improve text fidelity. Another closely related paper is Chen et al. (2019b), which predicts whether a statement align with records in the table. Since the nature of classification task makes it possible to model table records bidirectionally, it use BERT with templates to transform and model the table. Meanwhile, Chen et al. (2020a) explores coarse-to-fine table-to-text generation with standard GPT-2 model. Different from above two works, we adapted GPT-2 in a text generation scenario for structured data input and more importantly address table structure modeling and improve text fidelity. In addition, one of the auxiliary task: content matching is inspired by ideas in machine translation (Yang et al., 2019a) and Seq2Seq (Chen et al., 2019a) model. The closest paper on data-to-text generation is Wang et al. (2020). They assume that the expected generated text should cover all information in the table. But in a more realistic scenario, like the task we explore in this paper, the table consists of redundant information and only the important ones should be used to constraint model to generate high-fidelity text. Therefore, we propose to match important information only in the table and information in text as an auxiliary task during training.

6 Conclusion

In this work, we present TableGPT, which enhances GPT-2 for table-to-text generation with two auxiliary tasks, table structure reconstruction and content matching, for improving text fidelity while exploiting GPT-2's learnt linguistic knowledge from pretraining on large-scale corpus. In detail, we use table transformation to bridge the gap between structured table and natural language input for GPT-2 and further enhance GPT-2 with following two auxiliary tasks for table-to-text generation. The table structure reconstruction task help model preserve the structural information of input while representing table with powerful pretrained GPT-2. In addition, the content matching task guides model to generate high-fidelity

task with less incorrect expressions that are contradicting to the table via measuring distance between table and information in generated text. Experiments are conducted on three datasets, *Humans*, *Books* and *Songs*, in different domains. Both automatic evaluation and human evaluation results show that our model achieves new state-of-the-art performance in most few-shot setting.

Acknowledgements

References

Junwei Bao, Duyu Tang, Nan Duan, Zhao Yan, Yuanhua Lv, Ming Zhou, and Tiejun Zhao. 2018. Table-to-text: Describing table region with natural language. In *AAAI*, pages 5020–5027.

Liqun Chen, Yizhe Zhang, Ruiyi Zhang, Chenyang Tao, Zhe Gan, Haichao Zhang, Bai Li, Dinghan Shen, Changyou Chen, and Lawrence Carin. 2019a. Improving sequence-to-sequence learning via optimal transport. In *ICLR*.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2019b. Tabfact: A large-scale dataset for table-based fact verification. *ICLR*.

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020a. Logical natural language generation from open-domain tables. In *ACL*, pages 7929–7942.

Zhiyu Chen, Harini Eavani, Wenhu Chen, Yinyin Liu, and William Yang Wang. 2020b. Few-shot NLG with pre-trained language model. In *ACL*, 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 *NAACL-HLT*, pages 4171–4186.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the State-of-the-Art of End-to-End Natural Language Generation: The E2E NLG Challenge. *Computer Speech & Language*, 59:123–156.

Xiaocheng Feng, Yawei Sun, Bing Qin, Heng Gong, Yibo Sun, Wei Bi, Xiaojiang Liu, and Ting Liu. 2020. Learning to select bi-aspect information for document-scale text content manipulation. In *AAAI*, pages 7716–7723.

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 *EMNLP*, pages 3141–3150.

Sadid A Hasan and Oladimeji Farri. 2019. Clinical natural language processing with deep learning. In *Data Science for Healthcare*, pages 147–171.

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.

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*, pages 4881–4888.

Tianyu Liu, Fuli Luo, Qiaolin Xia, Shuming Ma, Baobao Chang, and Zhifang Sui. 2019. Hierarchical Encoder with Auxiliary Supervision for Neural Table-to-Text Generation: Learning Better Representation for Tables. In *AAAI*, pages 6786–6793.

Giulia Luise, Alessandro Rudi, Massimiliano Pontil, and Carlo Ciliberto. 2018. Differential properties of sinkhorn approximation for learning with wasserstein distance. In *Advances in Neural Information Processing Systems*, pages 5859–5870.

Hongyuan Mei, TTI UChicago, 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.

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.

Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. 2020. Few-shot natural language generation for task-oriented dialog. *arXiv preprint arXiv:2002.12328*.

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

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019b. Data-to-text generation with entity modeling. In *ACL*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Lei Sha, Lili Mou, Tianyu Liu, Pascal Poupart, Sujian Li, Baobao Chang, and Zhifang Sui. 2018. Order-planning neural text generation from structured data. In *AAAI*, pages 5414–5421.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing.

Mingming Yang, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, Min Zhang, and Tiejun Zhao. 2019a. Sentence-level agreement for neural machine translation. In *ACL*, pages 3076–3082.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019b. Xl-net: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *ACL*, pages 270–278.