Attend, Memorize and Generate: Towards Faithful Table-to-Text Generation in Few Shots

Wenting Zhao1, Ye Liu1, Yao Wan2, Philip S. Yu1

1 Department of Computer Science, University of Illinois at Chicago, IL, USA
2 School of Computer Sci. & Tech., Huazhong University of Science and Technology, China
{wzhao41, yliu279, psyu}@uic.edu, wanyao@hust.edu.cn

Abstract

Few-shot table-to-text generation is a task of composing fluent and faithful sentences to convey table content using limited data. Despite many efforts having been made towards generating impressive fluent sentences by fine-tuning powerful pre-trained language models, the faithfulness of generated content still needs to be improved. To this end, this paper proposes a novel approach Attend, Memorize and Generate (called AMG), inspired by the text generation process of humans. In particular, AMG (1) attends over the multi-granularity of context using a novel strategy based on table slot level and traditional token-by-token level attention to exploit both the table structure and natural linguistic information; (2) dynamically memorizes the table slot allocation states; and (3) generates faithful sentences according to both the context and memory allocation states. Comprehensive experiments with human evaluation on three domains (i.e., humans, songs, and books) of the Wiki dataset show that our model can generate higher quality texts when compared with several state-of-the-art baselines, in both fluency and faithfulness.1

1 Introduction

Table-to-text generation, which aims to translate a semi-structured table into natural language descriptions while preserving the conveyed table information, are drawing increasing interest over the past few years. It has been widely applied in many real-world scenarios, such as automatically generating weather forecasting reports (Liang et al., 2009), biographies (Lebret et al., 2016; Wang et al., 2018), task-oriented descriptions (Novikova et al., 2017), task-oriented conversations (Budzianowski et al., 2018; Williams et al., 2013) as well as healthcare descriptions (DiMarco et al., 2007; Hasan and

1All the source code and experimental dataset are available at https://github.com/wentinghome/AMG.

Figure 1: A motivating example.

Farri, 2019). Despite such significant gains, current approaches are driven by large-scale well-labeled training data, hindering the generalization to other scenarios with limited labeled data. In addition, the faithfulness of generated contents is still not well explored.

Few-shot natural language generation (Brown et al., 2020; Schick and Schütze, 2021; Xia et al., 2020a) has been in increasing demand since sufficient labeled data are always unavailable in many scenarios. To improve the table-to-text generation in few-shot scenarios, many existing works (Chen et al., 2020c; Gong et al., 2020; Peng et al., 2020) resort to the pre-training techniques which have been widely adopted in NLP, that is, pre-training a model first on large-scale unlabeled data, and then transfer the learned knowledge in pre-trained model to the few-shot scenario of table-to-text generation. Although these pre-trained models have achieved promising performance on generating fluent descriptions, from our investigation, they are still suffering from three major limitations: (1) The structure of table has not been well preserved. On table representation, existing methods (Chen et al., 2020c; Gong et al., 2020; Chen et al., 2020a) used to flatten the table into sequential sentences, ignoring the structured features (e.g., correlation between words within each table slot) among tables,
which is also critical for table-to-text generation. 

(2) **Generation bias.** Current approaches that directly fine-tune the model on target data make the model in favor of the knowledge learned from pre-training rather than specific target task knowledge, hurting the faithfulness because extra information irrelevant to the input table is introduced.

For example, as shown in Figure 1, given a table in the top box, the aim is to generate a coherent and faithful sentence with high coverage of table slots, as well as less out-of-table information. From this table, we can observe that current state-of-the-art models tend to generate sentences with hallucinated contents. For example, GPT-2 introduces wrong middle name “kelly” and the nationality “american”. In addition, the table coverage of contents generated by current approaches is low. For example, BART does not mention the event “marathon”. These observation motivate us to design a model that can generate faithful texts from tables while keeping the fluency.

To tackle the aforementioned limitations, this paper proposes a novel approach **Attend, Memorize and Generate** (called AMG) for faithful table-to-text generation in few-shots. Inspired by the human generation process which copies a consecutive slot span to compose a sentence using the context, we propose a table slot attention mechanism to empower the model generalization ability in inference by strengthening the dependency between the generated sentence with the input table. In addition, to avoid generating hallucinated contents, we design a memory unit to monitor the visits of each table slot. Particularly, the memory unit is initialized as all the meta-data of table slots, and then updated by checking the generated words as well as the current memory state.

Looking back to Figure 1, we can also observe several advantages of AMG. First of all, we can see AMG allows the to-be-predicted word “1998” from “birth_date” table slot to attend on the table as well as the previously generated sentence “robert ... born”, while the attention on within table slot words are prohibited. Thus, the model is enforced to capture the table span structure and rely on the table span value to generate. To this end, the model learns to capture the slot level table representation.

 Furthermore, as shown in Figure 1, “M₀” is the memory initial state where all the slot are available to be chosen (marked by green). After predicting the last word of table slot “name”, “M₁” will be updated since it detects that the table slot “name” is present in the generated sentence, thus making the state of “name” unavailable (marked by red). In addition, the generation of word “1998” takes the context and table slot allocation into account, therefore “1998” is selected by locating the value of table span “birth_date” as well as the activated signal of table slot “birth_date” (marked by blue) from memory allocation status.

To summarize, the primary contributions of this paper are as follows: (1) To better preserve the structure of table, we design a multi-grain attention that can attend over the table word as well as table slots level. (2) It is the first time that we introduce a memory mechanism to improve the faithfulness of generated texts by tracking the allocation of table slots. (3) We have conducted comprehensive experiments on three domains (i.e., Humans, Books and Songs) of the Wiki dataset to validate the effectiveness of our proposed approach.

## 2 Preliminaries

### 2.1 Problem Definition

Given a table $T$ of $m$ attribute-value pairs $\{(a_i, v_i)\}_{i=1}^m$, where $a_i$ and $v_i$ refer to the attribute name and value of $i$-th table slot, respectively, the table-to-text generation task aims at producing a coherent text $Y = (y_1, \cdots, y_L)$ that can describe the table information with fluency and faithfulness, where $L$ denotes the length of generated text.

### 2.2 UniLM

To alleviate the under-fitting issue caused by insufficient training examples in few shot learning, AMG adopts the state-of-art pre-trained language model UniLM (Dong et al., 2019) structure to integrate the external knowledge. UniLM is a multi-layer Transformer network which can be applied into both tasks of natural language understanding (NLU) and natural language generation (NLG). In this paper, we configure UniLM using Seq2Seq self-attention mask to aggregate the context of the masked $i$-th to-be-predicted word $y_i^{[MASK]}$ that are source sequence words from table $T$, and the previously generated target words $y_{<i}$. The proposed model computes the conditional probability for the to-be-predicted word using the masked language model objective function, as follows:

$$P(Y|T; \theta) = \prod_{i=1}^{L} P(y_i^{[MASK]}|y_{<i}, T; \theta). \quad (1)$$
3 AMG Approach

3.1 Overview
Figure 2 illustrates the overall architecture of our model, which is composed of three components, i.e., attend, memorize, and generate. (1) Attend. We propose a multi-granularity attention mechanism which attends over both token level and the table slot level to capture the linguistic knowledge as well as table structure information. We think that these knowledge can improve the faithfulness of generated texts. (2) Memory. We develop a memory to store and keep track of the table slot allocation status. (3) Generate. We take both the context representation and the table slot allocation states into account while making predictions. The above three building blocks interweave and lead the model to generate descriptions from tables faithfully.

3.2 Table Representation
Table Linearization Table-to-text generation receives semi-structured table as input. However, our proposed model AMG is built upon the UniLM architecture which requires natural sentence as input. Therefore, the first step we need to do is to translate the table into a natural sentence by linearization (Chen et al., 2020c). For the table example shown in Figure 1, the attribute value pair “name: robert kiprono cheruiyot” can be linearized as “name is [E_CLS] robert kiprono cheruiyot [E_SEP]”, where [E_CLS] and [E_SEP] are two special tokens to indicate the beginning and the end of table slot value.

Representing the History of Table Slot Allocation AMG makes prediction on the to-be-predicted token by taking the memory allocation status into account. The memory at different time step is updated by the previously generated table slots. Thus, we need to prepare the previously generated table slot representation $his^t$ at time step $t$ by using the static UniLM model. For example, in Figure 2, when making prediction for “[MASK]”, the representation of table slot allocation history is computed by feeding “robert kiprono cheruiyot” to the static UniLM model and obtain the average of hidden states.

3.3 Multi-Granularity Attention
AMG introduces the multi-granularity attention (MA) which is the combination of two granularity of attention, i.e., token level and table slot level attention. The token level attention is the original UniLM token level attention while the table slot level attention is the extra attention over table slot memory. The advantage is that the table slot attention can provide an extra signal to the UniLM, encouraging AMG to copy tokens from the table slot value that have not appeared in the target. As
shown in Figure 2, the memory augmented attention $A$ is the average of token level attention $A_{ta}$ and table slot level attention $A_{sa}$, as following:

$$A = (A_{ta} + A_{sa})/2,$$  \hspace{1cm} (2)

where the token level self-attention mechanism learns a unique series of query matrix $W_{Q_{ta}}^l$, key matrix $W_{K_{ta}}^l$, and value matrix $W_{V_{ta}}^l$ at the $l$-th Transformer layer for each attention head. Then, AMG maps the $(l-1)$-th Transformer layer output $T^{l-1}$ to three matrices: query $Q_{ta}$, key $K_{ta}$, and value $V_{ta}$. The output of a self-attention head $A_{ta}$ is computed as Eq.(3), where $Mask^{ta} \in \mathbb{R}^{N \times N}$ is the seq2seq attention mask, allowing the to-be-predicted token to attend to table tokens as well as the previously generated tokens. $N$ refers to the total token length of table, previously generated tokens and the current to-be-predicted token.

$$A_{ta}^{t} = \text{softmax}(\frac{Q_{ta}K_{ta}^T}{\sqrt{d_k}} + Mask^{ta}) \cdot V_{ta}, \hspace{1cm} (3)$$

**Table Slot Attention** Table slot attention works in a similar way with the self attention, while the major difference is to learn new key and value mapping matrices $W_{K_{sa}}^l$ and $W_{V_{sa}}^l$ and project memory $M^{l-1}$ using $W_{K_{sa}}^l$ and $W_{V_{sa}}^l$ to obtain $K_{sa}$ and $V_{sa}$. The query $Q_{sa}$ is computed by the projection of UniLM hidden state $h^{l-1}$ using mapping matrix $W_{Q_{sa}}^l$. Memory $M$ in AMG is defined as a $\mathbb{R}^{d_k \times slot_n}$ matrix where $slot_n$ is the maximum number of table slots. The $j$-th column of memory at time step $t$ is denoted as $M^l_j$, and the initial state of memory $M^0_j$ is the average embedding of the table slot value computed using static UniLM model. The output of slot level attention head $A_{sa}$ is as follows:

$$Q_{sa} = h^{l-1}W_{Q_{sa}}^l$$

$$K_{sa} = M^{l-1}W_{K_{sa}}^l$$

$$V_{sa} = M^{l-1}W_{V_{sa}}^l$$

$$A_{sa}^{t} = \text{softmax}(\frac{Q_{sa}K_{sa}^T}{\sqrt{d_k}} + Mask^{slot}) \cdot V_{sa}. \hspace{1cm} (4)$$

Instead of applying the original seq2seq attention from UniLM to the input, a table slot attention mask $Mask^{slot} \in \mathbb{R}^{N \times N}$ is introduced to decide which word should be attended. In our case, we prohibit the to-be-predicted token to attend the previously generated words within the same table slots, while allow to attend the rest of generated words and the table. As shown in Figure 2, “1998” from the descriptive sentence can attend to both the table “name is . . . , birth_date is . . .” and previously generated words “robert kiprono cheruiyot (born”, while is not allowed to attend to words within the same table slot “august 10.”.

**Table Slot Memory Update** AMG updates the memory matrix multiple times dynamically depending on how many times the generated sentence finishes generating one entire table slot value. To give a clear signal for the model to detect the beginning and the end of the table slot value, we introduce two additional special tokens \([E\_CLS]\) and \([E\_SEP]\) into the reference. Memory is updated using the gated mechanism, following (Henaff et al., 2016):

$$\hat{M}^t_j = \tanh(W_aM_j^{t-1} + W_bhis^{t-1})$$

$$z^t_j = \delta(W_cM_j^{t-1} + W_dhis^{t-1})$$

$$M^t_j = (1-z^t_j)M^{t-1} + z^t_j\hat{M}^t_j. \hspace{1cm} (5)$$

In Eq.(5), $W_a$, $W_b$, $W_c$ and $W_d$ are trainable parameters. First, $M^0_j$ is the new candidate memory to be combined with the existing memory $M^{t-1}_j$. Then, the gate function $z^t_j$ employs a sigmoid function $\delta$ to determine how much memory $M^t_j$ will be influenced. At last, we retain $M^t_j$ by using gate function to control how much each cell in memory is updated by considering the history of table slot appearance in the target sentence, as well as the last memory.

**Text Generation** When predicting the next token at each time step, AMG considers both the context representation and the table slot allocation status from memory shown in Eq.(6) where $tb$ refers to the table representation, $tk^t$ denotes the token predicted at time $t$ by AMG, and $tk^{0...t-1}$ denote the tokens previously generated from time 0 to $t - 1$.

$$(his^t, M^t, tk^t) = \text{AMG}(tb, his^{t-1}, M^{t-1}, tk^{0...t-1}). \hspace{1cm} (6)$$

### 3.4 Task-Adaptive Pre-Training

AMG is built upon the pre-trained UniLM and introduces additional weight. The memory updater depends on $W_a$, $W_b$, $W_c$ and $W_d$ to project memory and history values, as shown in Eq.(5). Besides, the newly added special token \([E\_CLS]\) and \([E\_SEP]\) is supposed to learn appropriate embedding weight from scratch. It is challenging to
<table>
<thead>
<tr>
<th></th>
<th>BLEU-4</th>
<th>METEOR</th>
<th>ROUGE-L</th>
<th>PARENT(P/R/F)</th>
<th>PARENT-T(P/R/F)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Humans</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>GPT2+copy (Chen et al., 2020c)</td>
<td>41.7</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>GPT2+copy (our replication)</td>
<td>42.05</td>
<td>33.36</td>
<td>63.90</td>
<td>68.47/37.28/45.59</td>
</tr>
<tr>
<td>3</td>
<td>TableGPT2 (Gong et al., 2020)</td>
<td>45.6</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>GPT2 (Radford et al., 2019)</td>
<td>24.26</td>
<td>25.20</td>
<td>53.90</td>
<td>74.04/41.46/50.79</td>
</tr>
<tr>
<td>5</td>
<td>BART (Lewis et al., 2020)</td>
<td>48.31</td>
<td>37.24</td>
<td>68.24</td>
<td>72.90/40.24/49.61</td>
</tr>
<tr>
<td>6</td>
<td>UniLM (Dong et al., 2019)</td>
<td>45.31</td>
<td>37.10</td>
<td>68.36</td>
<td>72.90/40.24/49.61</td>
</tr>
<tr>
<td>7</td>
<td>AMG</td>
<td>49.02</td>
<td>37.97</td>
<td>69.37</td>
<td>74.14/42.74/51.86</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Books</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>GPT2+copy (Chen et al., 2020c)</td>
<td>40.30</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>GPT2+copy (our replication)</td>
<td>40.39</td>
<td>34.48</td>
<td>67.59</td>
<td>69.68/35.10/44.87</td>
</tr>
<tr>
<td>3</td>
<td>TableGPT2 (Gong et al., 2020)</td>
<td>41.6</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>GPT2 (Radford et al., 2019)</td>
<td>19.12</td>
<td>24.99</td>
<td>54.83</td>
<td>72.86/37.84/48.11</td>
</tr>
<tr>
<td>5</td>
<td>BART (Lewis et al., 2020)</td>
<td>43.53</td>
<td>36.45</td>
<td>67.22</td>
<td>71.90/35.60/45.87</td>
</tr>
<tr>
<td>6</td>
<td>UniLM (Dong et al., 2019)</td>
<td>40.56</td>
<td>35.71</td>
<td>68.85</td>
<td>71.90/35.60/45.87</td>
</tr>
<tr>
<td>7</td>
<td>AMG</td>
<td>43.88</td>
<td>36.98</td>
<td>70.57</td>
<td>73.26/38.18/48.59</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Songs</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>GPT2+copy (Chen et al., 2020c)</td>
<td>42.20</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>GPT2+copy (our replication)</td>
<td>42.41</td>
<td>33.43</td>
<td>65.18</td>
<td>66.34/35.72/44.75</td>
</tr>
<tr>
<td>3</td>
<td>TableGPT2 (Gong et al., 2020)</td>
<td>42.30</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>GPT2 (Radford et al., 2019)</td>
<td>22.48</td>
<td>24.09</td>
<td>55.92</td>
<td>55.05/17.90/25.68</td>
</tr>
<tr>
<td>5</td>
<td>BART (Lewis et al., 2020)</td>
<td>43.88</td>
<td>34.69</td>
<td>67.22</td>
<td>69.22/36.31/46.00</td>
</tr>
<tr>
<td>6</td>
<td>UniLM (Dong et al., 2019)</td>
<td>42.63</td>
<td>34.79</td>
<td>67.92</td>
<td>68.19/34.74/44.55</td>
</tr>
<tr>
<td>7</td>
<td>AMG</td>
<td>45.09</td>
<td>35.55</td>
<td>67.38</td>
<td>67.60/37.63/46.90</td>
</tr>
</tbody>
</table>

Table 1: Test results on three domains Humans/Books/Songs of Wiki dataset using 500 training data. “P/R/F” denotes the precision/recall/F score.

We expect the newly introduced weight can be learned properly if we directly fine-tune AMG under the few shot scenario.

Inspired by the pre-trained language models and the task adaptive pre-training (Gururangan et al., 2020), we collect the unlabelled table side data to do a second phase task adaptive pre-training.

We first linearize the input table and add special token \[E_{CLS}\] and \[E_{SEP}\] to indicate the beginning and the end of the table slot value respectively. Then, around 20% tokens are masked and the cross entropy loss is employed as the objective function. One corrupted example for further pre-training stage is “\[CLS\] name is \[E_{CLS}\] [MASK] kiprono \[MASK\] \[E_{SEP}\]; birth_date is \[E_{CLS}\] 10 august \[MASK\] \[E_{SEP}\]... \[SEP\]”.

During pre-training, AMG modifies the UniLM model architecture by designing a novel slot attention mask as well as slot memory mechanism which introduces additional weights. There are two goals for pre-training: 1) tune UniLM weights to incorporate slot attention mask, and 2) learn proper weights for slot memory block. We divide the pre-training stage into two phases: slot attention based pre-training and slot memory based pre-training.

We incrementally incorporate the slot attention and slot memory elements to the UniLM model along the two pre-training phases. First, the model structure of slot attention based pre-training is to add the slot attention mask to the last 6 layers of UniLM. We also learn the embedding of two special tokens \[E_{CLS}\] and \[E_{SEP}\] by adding them into the UniLM vocabulary. We load the UniLM checkpoint model weight as the initial weight for slot attention based pre-training. The second slot memory based pre-training phase adopts the full AMG model, and is loaded with the checkpoint obtained after the slot attention mask based pre-training.

### 3.5 Fine-Tuning and Inference

In fine-tuning stage, AMG first loads the model weight after the further pre-training stage which exploits valuable information from plenty of unlabelled task relevant data. The input for our proposed model is the concatenation of the linearized table and the reference sentence. The model is trained end to end in masked language model fashion. Around 70% words in the reference are masked, and the cross entropy loss is used to minimize the discrepancy between the masked token
and the groundtruth.

For inference, table side data is present while the reference sentence is missing. Our approach generates sentence auto-regressively. When making prediction on the $t$-th word, we need to inform the model previously generated table slots through table slot history representation $his_t$.

4 Experiment

In this section, we explore the following experimental questions: (1) Can the proposed model generate fluent sentences?; and (2) Is the generated sentence faithful to the fact given by input table? We also perform ablation analysis to investigate the two main components of AMG, namely the slot attention and slot memory mechanism.

4.1 Dataset

**Task Adaptive Dataset for Pre-training** To pre-train AMG, we collect additional unlabelled data from WikiBio (Lebret et al., 2016) and Wiki dataset. First, Wiki-Humans is a subset of WikiBio dataset which contains massive training examples collected from Wikipedia, a cleaned-up version of original WikiBio dataset by setting a vocabulary bound and removing those include out-of-vocabulary words that are not in the given table. Since pre-training only requires the table side data and focuses on reconstructing the corrupted text, we collect the rest of table side data (around 500K from WikiBio by removing all the train/valid/test data used in Wiki-Humans heuristically. Second, for songs and books domain, we collect around 26K and 17K filtered out table data from (Chen et al., 2020c) respectively as the pre-training data.

**Dataset for Fine-Tuning** Inspired by the experimental settings of few-shot natural language generation in (Chen et al., 2020c), we conduct experiments on three domains, i.e., humans, songs and books of Wiki dataset denoted as Wiki-Humans, Wiki-Songs and Wiki-Books. For each domain, we fine-tune AMG to inspect the model performance on various few shot settings by sampling different amount of training examples (e.g. 500, 200, 100, 50). The validation set for each domain includes 1000 instances, and test sets of humans, songs and books domain have 13587, 11879 and 5252 examples. We set the maximum length of the linearized table and the generated sentence as 300 and 64 respectively.

4.2 Implementation Details

The base model for AMG is UniLM-base model with 12 Transformer layers, 768 hidden state dimensions, and 110M parameters in total. The implementation of AMG is divided into two stages in total: 1) two-phase task adaptive pre-training, and 2) fine-tuning on the target wiki dataset. We run the program on a single 1080Ti GPU with 12GB memory. Due to the memory constraint, the batch size on all stages is set as 4 and gradient is accumulated every 11 steps which results in a comparable 44 batch size. The learning rate is $5e-5$. The Adam (Kingma and Ba, 2015) optimizer is used and the weight decay is set as 0.01.

For fine-tuning, we fine-tune the AMG on target dataset by setting the maximum number of epoch as 50. For inference, we decode on the test set using the best checkpoints according to the validation set result. During inference, we use beam search with beam size 3 and length penalty 1.

4.3 Baselines

We compare the proposed model with strong pre-trained language models. UniLM (Dong et al., 2019) is a pre-trained language model for both natural language understanding and generation using three types of language modeling tasks. BART (Lewis et al., 2020) introduces a denoising autoencoder for pre-training sequence-to-sequence models. GPT-2 (Radford et al., 2019) is a powerful unidirectional model pre-trained on millions of webpages in auto-regressive fashion. GPT2+copy (Chen et al., 2020c) designed for few-shot table-to-text generation learns how to alternate between copying from table and generating functional words using GPT-2. TableGPT (Gong et al., 2020) is a followup work of (Chen et al., 2020c) while considers to minimize the contradicting part of the generated sentence given the table information.

4.4 Automatic Evaluation

Following other generation tasks, we choose three automatic evaluation metrics BLEU-4 (Papineni et al., 2002), ROUGE-L (Lin, 2004) and METEOR (Banerjee and Lavie, 2005) to evaluate the overlapping between the generated sentence and the reference sentence. Besides, to evaluate the faithfulness of generated sentence with the source table, we adopt PARENT (Dhingra et al., 2019) as our main metric. PARENT not only considers the
matching between the generated sentence with the reference, but also takes how much table slot information is reflected in the generated sentence into account. In addition, to further evaluate the faithfulness of the generated text, PARENT-T (Wang et al., 2020) which only measures the matching between the generated sentence and the corresponding table is also included.

Results  We first compare AMG with state-of-the-art models mentioned in section 4.3. Table 1 shows the performance of AMG and baseline models on three domains of Wiki dataset using 500 training examples. For (Chen et al., 2020c), we copy the code that the author released on GitHub and replicate the result denoted as GPT2+copy (our replication). Regarding the conventional overlapping based metrics BLEU-4, METEOR, ROUGE-L, We can see that AMG provides the best overall performance under various domains and evaluation metrics. AMG outperforms the base model UniLM 3.71%/3.32%/2.46% on BLEU-4 under Humans/Books/Songs domains, and AMG gains 0.73%/0.53%/0.16% more than the second best model BART on METEOR. AMG outperforms the second best model BART 1.07%/0.48%/0.90% on the F score of PARENT which is a strong indication that AMG can achieve the strongest balance between the fluency and faithfulness. Regarding the overlapping between the generated sentence with table content, F scores of PARENT-T metric shows that AMG provides the most informative results on Humans and Songs domains while still very competitive with the best model BART on Books domain.

Besides, to verify the stability of AMG when the amount of training data varies to 50, 100, 200 and 500, we show PARENT score for the proposed and other baseline models in Table 2. As shown in the table, over various domain and number of training example settings, AMG outperforms other baseline models. Specifically, under the 200 training examples, AMG outperforms the second strongest model BART by 2.72% on Humans, UniLM by 3.39% on Books, and BART by 1.81% on Songs. The results demonstrate that leveraging the table slot attention as well as the memory mechanism provide a stable and competitive performance of faithful generation. On the other hand, on the Humans/Books/Songs domain with 50 training examples, AMG gains 5.82%/1.74%/0.29% improvements than the second best model BART respectively which shows that our model has powerful generative ability even only 50 examples are present. And human domains achieves the most gain since we collect most pre-training data for the task adaptive pre-training, thus it would be beneficial for the further work to collect more task adaptive pre-training data for Books and Songs domains to further boost the model performance.

4.5 Analysis

We further analysis the faithfulness and the overall quality of the generated descriptions by conducting human evaluation. Then, we design ablation studies to investigate the importance of two building blocks of AMG: span attention and memory mechanism. In addition, we sample a specific input table and compare sentence generated by AMG with the state-of-the-art models shown in Figure 3.

Table 2: PARENT F score on three domains using 50/100/200/500 training examples.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Humans</th>
<th>Books</th>
<th>Songs</th>
</tr>
</thead>
<tbody>
<tr>
<td># of training examples</td>
<td>50</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>GPT2+copy (our replication)</td>
<td>30.59</td>
<td>34.59</td>
<td>40.54</td>
</tr>
<tr>
<td>GPT2 (Radford et al., 2019)</td>
<td>0.17</td>
<td>12.90</td>
<td>19.02</td>
</tr>
<tr>
<td>BART (Lewis et al., 2020)</td>
<td>37.73</td>
<td>41.37</td>
<td>47.41</td>
</tr>
<tr>
<td>UniLM (Dong et al., 2019)</td>
<td>35.80</td>
<td>41.83</td>
<td>46.08</td>
</tr>
<tr>
<td>AMG</td>
<td>43.55</td>
<td>47.72</td>
<td>50.13</td>
</tr>
</tbody>
</table>

Table 3: Results of human evaluation.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Reference</th>
<th>#sup</th>
<th>#con</th>
<th>overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain</td>
<td>Reference</td>
<td>#sup</td>
<td>#con</td>
<td>overall</td>
</tr>
<tr>
<td>Humans</td>
<td>3.87</td>
<td>1.71</td>
<td></td>
<td>3.55</td>
</tr>
<tr>
<td>Books</td>
<td>3.73</td>
<td>1.69</td>
<td></td>
<td>3.61</td>
</tr>
<tr>
<td>Songs</td>
<td>4.017</td>
<td>1.53</td>
<td></td>
<td>3.24</td>
</tr>
<tr>
<td>UniLM (Dong et al., 2019)</td>
<td>3.92</td>
<td>1.65</td>
<td>3.52</td>
<td></td>
</tr>
<tr>
<td>AMG</td>
<td>4.023</td>
<td>1.75</td>
<td></td>
<td>3.22</td>
</tr>
</tbody>
</table>

Table 4: Statistical significance on human evaluation.
Human Evaluation  Following (Wang et al., 2020; Chen et al., 2020c), we recruit three human annotators who pass the College English Test (CET-6) English test\(^2\) to judge the quality of the generated sentence. We sample 100 test tables and collect corresponding outputs from AMG, and baseline models. The sentences are randomly shuffled to reduce human variance. We provide instructions for human annotators to evaluate the sentence quality from two aspects: faithfulness and overall quality. First, for faithfulness, they are supposed to identify the number of entities mentioned in the sentence. Then, they need to compare the entities with ones from source table. Finally, they are supposed to report the number of fact supported and contradicted from the table respectively. Subsequently, we compute the average number of supported and unsupported entities denoted by \#sup and \#con in Table 3. The second study evaluates the overall quality of the generated sentence from their fluency, grammatical correctness, and the information consistency with the table. To compare the overall quality of various models, annotators rank the sentences generated using different models from 1 (best) to 6 (worst) by comparing the sentence. The “overall” column refers to the average ranking of the model. Table 3 shows that AMG generates better quality sentences compared with other models. Specifically, the outputs generated by AMG contains the most information supported by the table and the overall quality is ranked the first place. Although it shows the number unsupported by the table is higher than other models, the overall quality still outperforms other models.

The overall ranking in Table 3 between BART and AMG is quite close, thus we ask 3 human evaluators to rate the generated sentences from 3 criteria, and then calculate the statistical significance of the overall rating between BART and AMG. We randomly sample 50 sentences for 50 and 100 training examples in few-shot cases respectively. Three annotators are instructed to re-evaluate the overall sentence quality by rating them from 1 (worst) to 5 (best) by considering the following 3 criteria: (1) \#sup, (2) \#con (see Table 3), (3) naturalness and grammar correctness. The results are listed as follows.

As shown in Table 4, comparing BART with AMG, the p-value \(p = 0.002\) of Wilcoxon signed-rank tests shows at 95\% confidence level, AMG is statistically significant with BART when training examples are as scarce as 50. While at 75\% confidence level, AMG is statistically significant with BART when training examples increase to 500.

Ablation Study  We also conduct ablation studies to understand each component of the proposed model, including slot attention and slot memory mechanism. Table 5 provides the ablation results under different evaluation metrics. It shows that AMG can still outperform all these two variants over all, certifying the effectiveness of each designed component in our model and we demonstrate that incorporating table slot attention and memory mechanism with the pre-trained model UniLM can boost the model performance.

Case Study  Figure 3 provides a sample input table from test set along with various model outputs. The top box contains an input table while the bottom box includes model generations. In the bottom box, we leave the content supported by table as black, unsupported as light brown, and blue for the remaining words. We find that the output of pre-

---

\(^2\)A national English as a foreign language test in China.
trained baseline models suffer from the following problems: (1) repetition, e.g., BART fails to generate person name “wayne” correctly while repeats the last two letters as “waynene”, (2) hallucination, e.g., GPT-2 generates a middle name “wayne” which is out of table, and GPT2+copy attempts to copy the “office” slot but fail to copy the entire information by introducing unsupported information “the oak house” and “2003”. By contrast, AMG provides the highest table coverage while keeping the sentence fluent which demonstrates the table slot span attention and memory mechanism enables the model to copy from the table slot level correctly and enhance the generation faithfulness.

5 Related Work

Table-to-Text Generation Recent years have witnessed much success on representing the semi-structured tabular data and generating text to describe the table. From our investigation, most existing methods for table-to-text generation are based on the RNN-based encoder-decoder framework (Lebret et al., 2016; Liu et al., 2018; Wiseman et al., 2018; Ma et al., 2019; Liu et al., 2019a). Ma et al. (2019) extend the table-to-text generation to low-resource scenario and put forward a Transformer-based model. Of late, as the pre-training language model (e.g., BERT and GPT) has achieved significant successes in NLP, many works also propose to pre-train a model for table understanding. Yin et al. (2020) pre-train a model for jointly understanding of tabular data around textual descriptions on large-scale paired data. Herzig et al. (2020) extend the architecture of BERT to encode tables as input, and propose a weakly supervised pre-training model for question answering over tables. Kale (2020) investigate the performance of pre-trained T5 (Raffel et al., 2019) on multiple table-to-text tasks and provide a benchmark for the future research. To keep the faithfulness of table on generation, one related work to ours is (Wang et al., 2020), which introduces a new table-text optimal-transport matching loss and a table-text embedding similarity loss based on the Transformer model to enforce the faithfulness during text generation.

Pre-Trained Language Model Our work is also related to model pre-training for NLP, which has brought dramatic improvements on natural language understanding (Devlin et al., 2019; Liu et al., 2019c; Clark et al., 2020; Sun et al., 2019) and generation (Song et al., 2019; Dong et al., 2019; Liu et al., 2020b, 2019b). The widely used pre-trained models (PTMs) for table-to-text generation can be categorized into two classes: text-to-text PTMs (Radford et al., 2018; Devlin et al., 2019; Dong et al., 2019; Lewis et al., 2020; Joshi et al., 2020) and structured data-to-text PTMs (Chen et al., 2020b; Herzig et al., 2020; Xing and Wan, 2021). Recently, many pre-training models (Liu et al., 2021, 2020a; Yao et al., 2019) start to incorporated the structured information from knowledge bases (KBs) or other structured semantic annotations into pre-training, which is also related to our work.

Few-shot text generation Few-shot text generation learns with minimal data while maintaining decent generation capacity. Few-shot text generation can be used to augment the scarce training data to better assist the down-stream task, e.g., (Xia et al., 2020a,b) for spoken language intent detection, (Bražinskas et al., 2020) for opinion summary generation. In addition, to better utilize the available resources, Chang et al. (2021) investigates the training instance selection on unlabelled data, and (Schick and Schütze, 2020) adapts pattern-exploiting training strategy to fine-tune a PTM.

6 Conclusion

In this paper, we have proposed a novel approach AMG for faithful table-to-text generation in few shots. We first attend over the multi-granularity of context using a novel span level and traditional token-by-token level attention strategy to exploit both the table structural and natural linguistic information. Then, we design a memory unit to memorize the table slot allocation states dynamically. Extensive experiments on three domains of Wiki dataset verify the effectiveness of our proposed model on generating fluent and faithful descriptions from tables.

Acknowledgements

We would like to thank all the anonymous reviewers for their helpful comments. This work is supported by NSF under grants III-1763325, III-1909323, III-2106758, and SaTC-1930941. Yao Wan is partially supported by the Fundamental Research Funds for the Central Universities.
References


