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

Wenting Zhao[1] **Ye Liu**[1] **Yao Wan**[2] **Philip S. Yu**[1]
[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 qualified 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), restaurant 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 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.

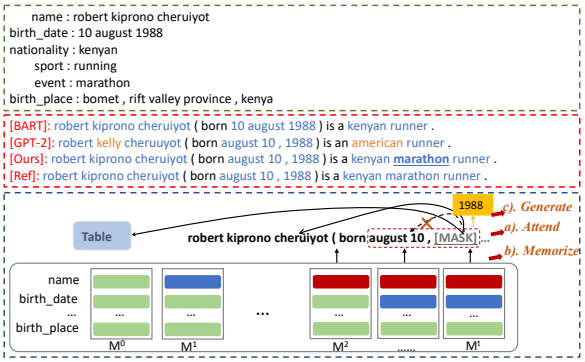

Figure 1: A motivating example.

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,

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

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 4106–4117
November 7–11, 2021.

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, "M^0" 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*", "M^1" 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:

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

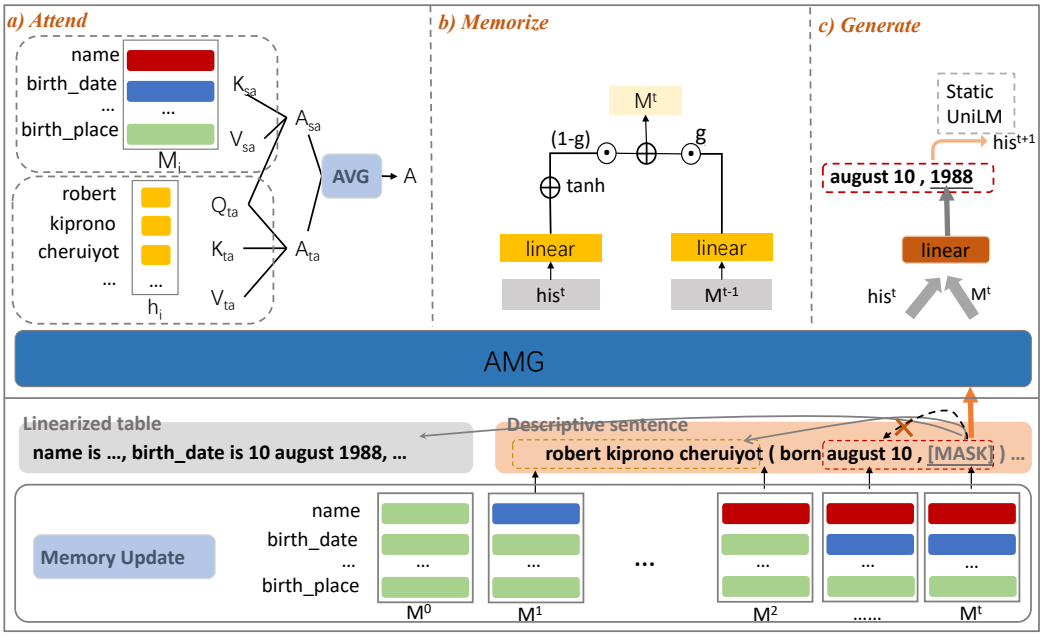

Figure 2: An overview of AMG. The input to AMG is the concatenation of linearized table (marked in grey) and the descriptive sentence(marked in orange). The bottom box shows the memory update process. The top three boxes show the building blocks of AMG, designed to attend, memorize and generate descriptions from tables.

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:

$$A = (A_{ta} + A_{sa})/2\,, \tag{2}$$

where the token level self-attention mechanism learns a unique series of query matrix $W^l_{Q_{ta}}$, key matrix $W^l_{K_{ta}}$, and value matrix $W^l_{V_{ta}}$ 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.

$$A^l_{ta} = \text{softmax}(\frac{Q_{ta}K^T_{ta}}{\sqrt{d^k}} + Mask^{ta}) \cdot V_{ta}\,, \tag{3}$$

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 $W^l_{K_{sa}}$ and $W^l_{V_{sa}}$ and project memory M^{l-1} using $W^l_{K_{sa}}$ and $W^l_{V_{sa}}$ 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 $W^l_{Q_{sa}}$. Memory M in AMG is defined as a $\mathbb{R}^{d_h \times slot_n}$ 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^t_j, and the initial state of memory M^0_j is the average embedding of the j-th table slot value computed using static UniLM model. The output of slot level attention head A^l_{sa} is as follows:

$$\begin{aligned}
Q_{sa} &= h^{l-1}W^l_{Q_{sa}} \\
K_{sa} &= M^{l-1}W^l_{K_{sa}} \\
V_{sa} &= M^{l-1}W^l_{V_{sa}} \\
A^l_{sa} &= \text{softmax}(\frac{Q_{sa}K^T_{sa}}{\sqrt{d^k}} + Mask^{slot}) \cdot V_{sa}\,.
\end{aligned} \tag{4}$$

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):

$$\begin{aligned}
\hat{M}^t_j &= \tanh(W_a M^{t-1}_j + W_b his^{t-1}) \\
z^t_j &= \delta(W_c M^{t-1}_j + W_d his^{t-1}) \\
M^t_j &= (1 - z^t_j)M^{t-1}_j + z^t_j \hat{M}^t_j\,.
\end{aligned} \tag{5}$$

In Eq.(5), W_a, W_b, W_c and W_d are trainable parameters. First, $\hat{M}^t_j$ 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 δ 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 $tk^{0\ldots t-1}$ denote the tokens previously generated from time 0 to $t-1$.

$$\begin{aligned}
&(his^t, M^t, tk^t) = \\
&\qquad \text{AMG}(tb, his^{t-1}, M^{t-1}, tk^{0\ldots t-1})\,.
\end{aligned} \tag{6}$$

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

		BLEU-4	METEOR	ROUGE-L	PARENT(P/R/F)	PARENT-T(P/R/F)
	Humans					
1	GPT2+copy (Chen et al., 2020c)	41.7	-	-	-	-
2	GPT2+copy (our replication)	42.05	33.36	63.90	68.47/37.28/45.59	47.90/40.18/41.58
3	TableGPT2 (Gong et al., 2020)	45.6	-	-	-	-
4	GPT2 (Radford et al., 2019)	24.26	25.20	53.90	59.45/18.51/25.89	41.60/27.93/31.57
5	BART (Lewis et al., 2020)	48.31	37.24	68.24	74.04/41.46/50.79	**51.50**/41.98/44.20
6	UniLM (Dong et al., 2019)	45.31	37.10	68.36	72.90/40.24/49.61	50.06/41.67/43.46
7	AMG	**49.02**	**37.97**	**69.37**	**74.14/42.74/51.86**	51.20/**43.03/44.70**
	Books					
1	GPT2+copy (Chen et al., 2020c)	40.30	-	-	-	-
2	GPT2+copy (our replication)	40.39	34.48	67.59	69.68/35.10/44.87	51.34/35.34/40.45
3	TableGPT2 (Gong et al., 2020)	41.6	-	-	-	-
4	GPT2 (Radford et al., 2019)	19.12	24.99	54.83	55.22/17.72/24.94	40.41/28.21/32.14
5	BART (Lewis et al., 2020)	43.53	36.45	68.93	72.86/37.84/48.11	**54.35/37.51/42.97**
6	UniLM (Dong et al., 2019)	40.56	35.71	68.85	71.90/35.60/45.87	53.07/35.58/41.15
7	AMG	**43.88**	**36.98**	**70.57**	**73.26/38.18/48.59**	53.89/37.29/42.69
	Songs					
1	GPT2+copy (Chen et al., 2020c)	42.20	-	-	-	-
2	GPT2+copy (our replication)	42.41	33.43	65.18	66.34/35.72/44.75	42.05/33.99/36.27
3	TableGPT2 (Gong et al., 2020)	42.30	-	-	-	-
4	GPT2 (Radford et al., 2019)	22.48	24.09	55.92	55.05/17.90/25.65	30.96/21.53/24.42
5	BART (Lewis et al., 2020)	43.88	34.69	67.22	**69.22**/36.31/46.00	**43.48**/34.55/37.26
6	UniLM (Dong et al., 2019)	42.63	34.79	**67.92**	68.19/34.74/44.55	41.32/32.64/35.24
7	AMG	**45.09**	**35.55**	67.38	67.60/**37.63/46.90**	42.78/**35.21/37.36**

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.

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 adpative 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 give 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

Domain	Humans				Books				Songs			
# of training examples	50	100	200	500	50	100	200	500	50	100	200	500
GPT2+copy (our replication)	30.59	34.59	40.54	45.59	42.67	42.79	43.44	44.87	40.18	41.72	43.97	44.75
GPT2 (Radford et al., 2019)	0.17	12.90	19.02	25.89	0.71	20.82	24.18	24.94	0.85	17.08	24.72	25.65
BART (Lewis et al., 2020)	37.73	41.37	47.41	45.45	41.68	43.43	43.65	48.11	41.74	42.44	44.12	46.00
UniLM (Dong et al., 2019)	35.80	41.83	46.08	49.61	38.28	41.39	44.06	45.87	40.17	41.95	42.45	44.55
AMG	**43.55**	**47.72**	**50.13**	**51.86**	**43.42**	**46.03**	**47.45**	**48.59**	**42.03**	**43.30**	**45.93**	**46.90**

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

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 text and the corresponding table is also included.

Domain	#sup	#con	overall
Reference	3.87	1.71	3.55
GPT2+copy (our replication)	3.99	1.75	3.39
GPT2 (Radford et al., 2019)	3.73	1.69	3.61
BART (Lewis et al., 2020)	4.017	**1.53**	3.24
UniLM (Dong et al., 2019)	3.92	1.65	3.52
AMG	**4.023**	1.75	**3.22**

Table 3: Results of human evaluation.

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 PARTENT-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.

	BART	AMG	
50 shots rating	3.87	4.11	$p = 0.002$
500 shots rating	4.46	4.55	$p = 0.24$

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.

[2]A national English as a foreign language test in China.

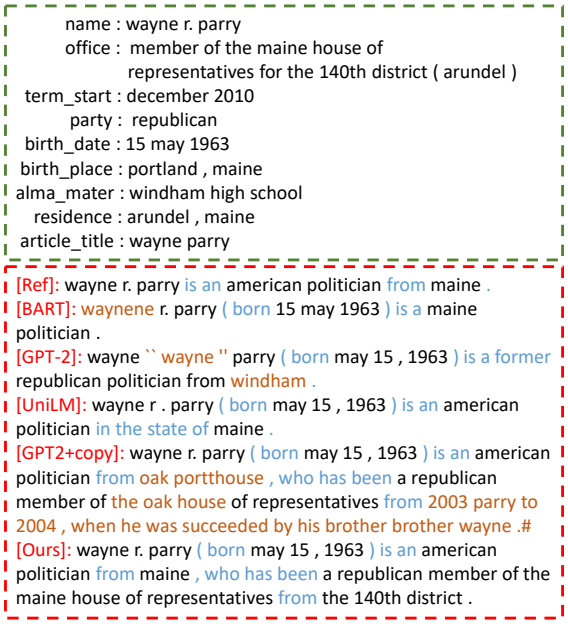

Figure 3: A case study of a specific table input for qualitative analysis of table-to-text generation.

Model	BLEU	METEOR	PARENT	PARENT-T
AMG	**49.02**	37.97	**51.86**	**44.70**
AMG w/o span	47.28	37.10	50.24	43.36
AMG w/o mem	48.92	**38.14**	51.38	43.76
AMG w/o extra	46.78	36.99	49.83	44.00

Table 5: Ablation study of the proposed model.

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 overall, 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-

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 ... brotherwayne.*". 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

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2020. Few-shot learning for opinion summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4119–4135, Online. Association for Computational Linguistics.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Ultes Stefan, Ramadan Osman, and Milica Gašić. 2018. Multiwoz - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ernie Chang, Xiaoyu Shen, Hui-Syuan Yeh, and Vera Demberg. 2021. On training instance selection for few-shot neural text generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 8–13, Online. Association for Computational Linguistics.

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020a. Logical natural language generation from open-domain tables. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7929–7942, Online. Association for Computational Linguistics.

Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. 2020b. KGPT: Knowledge-grounded pre-training for data-to-text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8635–8648, Online. Association for Computational Linguistics.

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

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William Cohen. 2019. Handling divergent reference texts when evaluating table-to-text generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4884–4895, Florence, Italy. Association for Computational Linguistics.

Chrysanne DiMarco, H Dominic Covvey, Peter Bray, Donald Cowan, Vic DiCiccio, Eduard Hovy, Joan Lipa, and Doug Mulholland. 2007. The development of a natural language generation system for personalized e-health information. *Medinfo*, 2007:12th.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

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 *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1978–1988, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

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

Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. 2016. Tracking the world state with recurrent entity networks. *arXiv preprint arXiv:1612.03969*.

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

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Mihir Kale. 2020. Text-to-text pre-training for data-to-text tasks. *arXiv preprint arXiv:2005.10433*.

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

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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

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

Tianyu Liu, Fuli Luo, Qiaolin Xia, Shuming Ma, Baobao Chang, and Zhifang Sui. 2019a. Hierarchical encoder with auxiliary supervision for neural table-to-text generation: Learning better representation for tables. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6786–6793.

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018. Table-to-text generation by structure-aware seq2seq learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020a. K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2901–2908.

Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip S Yu. 2021. Kg-bart: Knowledge graph-augmented bart for generative commonsense reasoning. In *Proceedings of the AAAI Conference onArtificial Intelligence*.

Ye Liu, Tao Yang, Zeyu You, Wei Fan, and Philip S Yu. 2020b. Commonsense evidence generation and injection in reading comprehension. In *Proceedings of SIGDIAL*.

Ye Liu, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S Yu. 2019b. Generative question refinement with deep reinforcement learning in retrieval-based qa system. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1643–1652.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019c. Roberta: A robustly optimized bert pretraining approach.

Shuming Ma, Pengcheng Yang, Tianyu Liu, Peng Li, Jie Zhou, and Xu Sun. 2019. Key fact as pivot: A two-stage model for low resource table-to-text generation. *arXiv preprint arXiv:1908.03067*.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206, Saarbrücken, Germany. Association for Computational Linguistics.

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

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. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 172–182, Online. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2020. Few-shot text generation with pattern-exploiting training.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

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

Zhenyi Wang, Xiaoyang Wang, Bang An, Dong Yu, and Changyou Chen. 2020. Towards faithful neural table-to-text generation with content-matching constraints. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1072–1086, Online. Association for Computational Linguistics.

Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 404–413, Metz, France. Association for Computational Linguistics.

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

Congying Xia, Caiming Xiong, Philip Yu, and Richard Socher. 2020a. Composed variational natural language generation for few-shot intents. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3379–3388, Online. Association for Computational Linguistics.

Congying Xia, Chenwei Zhang, Hoang Nguyen, Jiawei Zhang, and Philip Yu. 2020b. Cg-bert: Conditional text generation with bert for generalized few-shot intent detection. *arXiv preprint arXiv:2004.01881*.

Xinyu Xing and Xiaojun Wan. 2021. Structure-aware pre-training for table-to-text generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2273–2278, Online. Association for Computational Linguistics.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.

Pengcheng Yin, Graham Neubig, Wen tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. In *Annual Conference of the Association for Computational Linguistics (ACL)*.