Few-Shot NLG with Pre-Trained Language Model

Zhiyu Chen[1], **Harini Eavani**[2], **Wenhu Chen**[1], **Yinyin Liu**[2], and **William Yang Wang**[1]

[1]University of California, Santa Barbara
[2]Intel AI
{zhiyuchen, wenhuchen, william}@cs.ucsb.edu, {harini.eavani, yinyin.liu}@intel.com

Abstract

Neural-based end-to-end approaches to natural language generation (NLG) from structured data or knowledge are data-hungry, making their adoption for real-world applications difficult with limited data. In this work, we propose the new task of *few-shot natural language generation*. Motivated by how humans tend to summarize tabular data, we propose a simple yet effective approach and show that it not only demonstrates strong performance but also provides good generalization across domains. The design of the model architecture is based on two aspects: content selection from input data and language modeling to compose coherent sentences, which can be acquired from prior knowledge. With just 200 training examples, across multiple domains, we show that our approach achieves very reasonable performances and outperforms the strongest baseline by an average of over 8.0 BLEU points improvement. Our code and data can be found at https://github.com/czyssrs/Few-Shot-NLG

1 Introduction

Natural language generation (NLG) from structured data or knowledge (Gatt and Krahmer, 2018) is an important research problem for various NLP applications. Some examples are task-oriented dialog, question answering (He et al., 2017; Ghazvininejad et al., 2018; Su et al., 2016; Saha et al., 2018; Yin et al., 2016) and interdisciplinary applications such as medicine (Hasan and Farri, 2019; Cawsey et al., 1997) and healthcare (Hasan and Farri, 2019; DiMarco et al., 2007). There is great potential to use automatic NLG systems in a wide range of real-life applications. Recently, deep neural network based NLG systems have been developed, such as those seen in the E2E challenge (Novikova et al., 2017), WEATHERGOV (Liang et al., 2009), as well as more complex ones such as WIKIBIO (Liu et al., 2018) and ROTOWIRE (Wiseman et al., 2017). Compared to traditional slot-filling pipeline approaches, such neural-based systems greatly reduce feature engineering efforts and improve text diversity as well as fluency.

Although they achieve good performance on benchmarks such as E2E challenge (Novikova et al., 2017) and WIKIBIO (Lebret et al., 2016), their performance depends on large training datasets, e.g., 500k table-text training pairs for WIKIBIO (Lebret et al., 2016) in a single domain. Such data-hungry nature makes neural-based NLG systems difficult to be widely adopted in real-world applications as they have significant manual data curation overhead. This leads us to formulate an interesting research question:

> *1. Can we significantly reduce human annotation effort to achieve reasonable performance using neural NLG models?*
> *2. Can we make the best of generative pre-training, as prior knowledge, to generate text from structured data?*

Motivated by this, we propose the new task of *few-shot natural language generation*: given only a handful of labeled instances (e.g., 50 - 200 training instances), the system is required to produce satisfactory text outputs (e.g., BLEU $\geq$ 20). To the best of our knowledge, such a problem in NLG community still remains under-explored. Herein, we propose a simple yet very effective approach that can generalize across different domains.

In general, to describe information in a table, we need two skills to compose coherent and faithful sentences. One skill is to select and copy factual content from the table - this can be learned quickly by reading a handful of tables. The other is to compose grammatically correct sentences that bring those facts together - this skill is not re-

Input Table

Attribute (R)	Value (V)
Name	Walter Extra
Nationality	German
Occupation	Aircraft designer and manufacturer
...	...

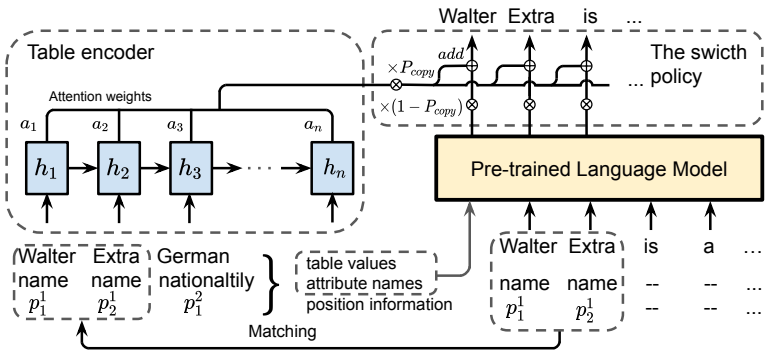

Figure 1: Overview of our approach: Under the base framework with switch policy, the pre-trained language model serves as the generator. We follow the same encoder as in (Liu et al., 2018). The architecture is simple in terms of both implementation and parameter space that needs to be learned from scratch, which should not be large given the few-shot learning setting.

stricted to any domain. One can think of a latent "switch" that helps us alternate between these two skills to produce factually correct and coherent sentences. To do this, we use the pre-trained language model (Chelba et al., 2013; Radford et al., 2019) as the innate language skill, which provides strong prior knowledge on how to compose fluent and coherent sentences. The ability to switch and select/copy from tables can be learned successfully using only a few training instances, freeing the neural NLG model from data-intensive training. Previous best performing methods based on large training data, such as (Liu et al., 2018), which does not apply such switch mechanism but trains a strong domain-specific language model, perform very poorly under few-shot setting.

Since we are operating under a highly data-restricted few-shot regime, we strive for simplicity of model architecture. This simplicity also implies better generalizability and reproducibility for real-world applications. We crawl multi-domain table-to-text data from Wikipedia as our training/test instances. With just 200 training instances, our method can achieve very reasonable performance.

In a nutshell, our contributions are summarized as the following:

- We propose the new research problem of few-shot NLG, which has great potential to benefit a wide range of real-world applications.
- To study different algorithms for our proposed problem, we create a multi-domain table-to-text dataset.
- Our proposed algorithm can make use of the external resources as prior knowledge to significantly decrease human annotation effort and improve the baseline performance by an average of over 8.0 BLEU on various domains.

2 Related Work

2.1 NLG from Structured Data

As it is a core objective in many NLP applications, natural language generation from structured data/-knowledge (NLG) has been studied for many years. Early traditional NLG systems follow the pipeline paradigm that explicitly divides generation into content selection, macro/micro planning and surface realization (Reiter and Dale, 1997). Such a pipeline paradigm largely relies on templates and hand-engineered features. Many works have been proposed to tackle the individual modules, such as (Liang et al., 2009; Walker et al., 2001; Lu et al., 2009). Later works (Konstas and Lapata, 2012, 2013) investigated modeling context selection and surface realization in an unified framework.

Most recently, with the success of deep neural networks, data-driven, neural based approaches have been used, including the end-to-end methods that jointly model context selection and surface realization (Liu et al., 2018; Wiseman et al., 2018; Puduppully et al., 2018). Such data-driven approaches achieve good performance on several benchmarks like E2E challenge (Novikova et al., 2017), WebNLG challenge (Gardent et al., 2017) and WIKIBIO (Lebret et al., 2016). However, they rely on massive amount of training data. ElSahar et al. (2018) propose zero-shot learning for question generation from knowledge graphs, but their work applies on the transfer learning setting for unseen knowledge base types, based on seen ones and their textual contexts, which still requires large in-domain training dataset. This is different from our few-shot learning setting. Ma et al. (2019) propose low-resource table-to-text generation with

1,000 paired examples and large-scale target-side examples. In contrast, in our setting, only tens to hundreds of paired training examples are required, meanwhile without the need for any target examples. This is especially important for real-world use cases where such large target-side gold references are mostly hard to obtain. Therefore, our task is more challenging and closer to real-world settings.

2.2 Large Scale Pre-Trained Models

Many of the current best-performing methods for various NLP tasks adopt a combination of pre-training followed by supervised fine-tuning, using task-specific data. Different levels of pre-training include word embeddings (Mikolov et al., 2013; Pennington et al., 2014; Peters et al., 2018), sentence embeddings (Le and Mikolov, 2014; Kiros et al., 2015), and most recently, language modeling based pre-training like BERT (Devlin et al., 2018) and GPT-2 (Radford et al., 2019). Such models are pre-trained on large-scale open-domain corpora, and provide down-streaming tasks with rich prior knowledge while boosting their performance. In this paper, we adopt the idea of employing a pre-trained language model to endow in-domain NLG models with language modeling ability, which cannot be well learned from few shot training instances.

3 Method

3.1 Problem Formulation

We are provided with semi-structured data: a table of attribute-value pairs $\{R_i : V_i\}_{i=1}^{n}$. Both R_i and V_i can be either a string/number, a phrase or a sentence. Each value is represented as a sequence of words $V_i = \{v_j\}_{j=1}^{m}$. For each word v_j, we have its corresponding attribute name R_i and position information of the word in the value sequence. The target is to generate a natural language description based on the semi-structured data, provided with only a handful of training instances.

3.2 Base Framework with Switch Policy

We start with the field-gated dual attention model proposed in (Liu et al., 2018), which achieves state-of-the-art performance (BLEU) on WIKIBIO dataset. Their method uses an LSTM decoder with dual attention weights. We first apply a switch policy that decouples the framework into table content selection/copying and language model based generation. Inspired by the pointer generator (See et al., 2017), at each time step, we maintain a soft switch p_{copy} to choose between generating from softmax over vocabulary or copying from input table values with the attention weights as the probability distribution.

$$p_{copy} = \text{sigmoid}(W_c c_t + W_s s_t + W_x x_t + b)$$

Where $c_t = \sum_i a_t^i h_i$, $\{h_i\}$ is the encoder hidden states, x_t, s_t, a_t is the decoder input, state and attention weights respectively at time step t. W_c, W_s, W_x and b are trainable parameters.

The pointer generator learns to alternate between copying and generating based on large training data and shows its advantage of copying out-of-vocabulary words from input. In our task, the training data is very limited, and many of the table values are not OOV. We need to explicitly "teach" the model where to copy and where to generate. Therefore, to provide the model accurate guidance of the behavior of the switch, we match the target text with input table values to get the positions of where to copy. At these positions, we maximize the copy probability p_{copy} via an additional loss term. Our loss function:

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

Where L_c is the original loss between model outputs and target texts. w_j is the target token at position j, $\{V_i\}$ is the input table value list defined in Section 3.1, and m means a matched phrase. λ is hyperparameter as the weight for this copy loss term. We also concatenate the decoder input with its matched attribute name and position information in the input table as x_t to calculate p_{copy} .

3.3 Pre-Trained LM as Generator

We use a pre-trained language model as the generator, serving as the "innate language skill". Due to the vocabulary limitation of few training instances, we leave the pre-trained word embedding fixed while fine-tuning other parameters of the pre-trained language model, so that it can generalize with tokens unseen during training.

Figure 1 shows our model architecture. We use the pre-trained language model GPT-2[1] proposed in (Radford et al., 2019), which is a 12-layer transformer. The final hidden state of the transformer is used to calculate attention weights and the copy

[1] https://github.com/openai/gpt-2

Domain	Humans					Books					Songs				
# of training instances	-	50	100	200	500	-	50	100	200	500	-	50	100	200	500
Template	16.3	-	-	-	-	25.6	-	-	-	-	30.1	-	-	-	-
Base-original	-	2.2	3.7	4.9	5.1	-	5.8	6.1	7.4	6.7	-	9.2	10.7	11.1	11.3
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	-	15.6	17.8	21.3	26.2	-	24.7	26.9	30.5	33.2	-	29.7	30.6	32.5	34.9
Base + switch + LM-scratch	-	6.6	11.5	15.3	18.6	-	7.1	9.2	14.9	21.8	-	11.6	16.2	20.6	23.7
Base + switch + LM (Ours)	-	**25.7**	**29.5**	**36.1**	**41.7**	-	**34.3**	**36.2**	**37.9**	**40.3**	-	**36.1**	**37.2**	**39.4**	**42.2**

Table 1: BLEU-4 results on three domains. Base-original: the original method in (Liu et al., 2018); Base: applies pre-trained word embedding; Base+switch: adds the switch policy; Base+switch+LM-scratch: makes the same architecture as our method, but trains the model from scratch without pre-trained weights for the generator. Template: manually crafted templates

switch p_{copy}. We first feed the embedded attribute-value list serving as the context for generation. In this architecture, the generator is fine-tuned from pre-trained parameters while the encoder and attention part is learned from scratch, the initial geometry of the two sides are different. Therefore we need to apply larger weight to the copy loss p_{copy}, to give the model a stronger signal to "teach" it to copy facts from the input table.

4 Experiment

4.1 Datasets and Experiment Setup

The original WIKIBIO dataset (Lebret et al., 2016) contains 700k English Wikipedia articles of well-known humans, with the Wiki infobox serving as input structured data and the first sentence of the article serving as target text. To demonstrate generalizability, we collect datasets from two new domains: *Books* and *Songs* by crawling Wikipedia pages. After filtering and cleanup, we end up with 23,651 instances for *Books* domain and 39,450 instances for *Songs* domain[2]. Together with the *Humans* domain of the original WIKIBIO dataset, for all three domains we conduct experiments by varying the training dataset size to 50, 100, 200 and 500. The rest of data is used for validation (1,000) and testing. The weight λ of the copy loss term is set to 0.7. Other parameter settings can be found in Appendix A. To deal with vocabulary limitation of few-shot training, for all models we adopt the Byte Pair Encoding (BPE) (Sennrich et al., 2016) and subword vocabulary in (Radford et al., 2019).

We compare the proposed method with other approaches investigated in Section 3, serving as the baselines - **Base-original:** the original model in (Liu et al., 2018); **Base:** uses the same architecture, but in addition applies the pre-trained word embedding and fix it during training; **Base + switch:** adds the switch policy; **Base + switch + LM-scratch:** makes the architecture same as our method, except training the model from scratch instead of using pre-trained weights for generator. **Template:** template-based non-neural approach, manually crafted for each domain.

[2]Note that the target text sometimes contains information not in the infobox. This is out of the scope of the few-shot generation in this work. Therefore we further filter the datasets and remove the ones with rare words out of infobox. Check (Dhingra et al., 2019) for a related study of this issue on the WikiBio dataset

4.2 Results and Analysis

Following previous work (Liu et al., 2018), we first conduct automatic evaluations using BLEU-4, shown in Table 1. The ROUGE-4 (F-measure) results follow the same trend with BLEU-4 results, which we show in Appendix B.

As we can see, the original model **Base-original** (Liu et al., 2018), which obtains the state-of-the-art result on WIKIBIO full set, performs very poorly under few-shot setting. It generates all tokens from softmax over vocabulary, which results in severe overfitting with limited training data, and the results are far behind the template-based baseline. With the switch policy, **Base+switch** first brings an improvement of an average of over 10.0 BLEU points. This indicates that the content selection ability is easier to be learned with a handful of training instances. However, it forms very limited, not fluent sentences. With the augmentation of the pre-trained language model, our model **Base+switch+LM** brings one more significant improvement of an average over 8.0 BLEU points. We provide sample outputs of these methods using 200 training instances in Table 2.

Table 3 shows the effect of the copy switch loss p_{copy} introduced in Section 3.2, giving the model a stronger signal to learn to copy from input table.

Ma et al. (2019) propose the Pivot model, for low-resource NLG with 1,000 paired examples and large-scale target-side examples. We compare our

Attribute	Value	Attribute	Value
name	andri ibo	fullname	andri ibo
birth date	3 april 1990	birth place	sentani , jayapura , indonesia
height	173 cm	currentclub	persipura jayapura
position	defender	...	
Gold Reference: andri ibo (born april 3 , 1990) is an indonesian footballer who currently plays for persipura jayapura in the indonesia super league .			
Generated texts of different methods			
Base: vasco emanuel freitas (born december 20 , 1992 in kong kong) is a hong kussian football player and currently plays for hong kong first division league side tsw pegasus .			
Base+switch: andri ibo andri ibo (3 april 1990) is a international cricketer .			
Base+switch+LM (Ours): andri ibo (born 3 april 1990) is an indonesian football defender , who currently plays for persipura jayapura .			

Table 2: A sample input table and generated summaries from the test set of *Humans* domain, using 200 training instances

# of training instances	50	100	200	500
Base + switch + LM	**25.7**	**29.5**	**36.1**	**41.7**
- w/o copy loss p_{copy}	21.4	25.5	31.3	38.0

Table 3: Ablation study: Effect of the copy loss term on *Humans* domain, measured by BLEU-4. The loss term brings an average improvement of over 4.0 BLEU points.

method with the Pivot model in table 4. Note that here we train and evaluate the models on the original WikiBio dataset used in their work, in order to maintain the size of the target side examples for their settings.

# of paired training instances	50	100	200	500	1000
Pivot	7.0	10.2	16.8	20.3	27.3
Ours	17.2	23.8	25.4	28.6	31.2

Table 4: Comparison with the Pivot model (Ma et al., 2019). Compared to their method using additional large-scale target side examples, our method requires no additional target side data, while achieving better performance.

Human Evaluation

We also conduct human evaluation studies using Amazon Mechanical Turk, based on two aspects: *Factual correctness* and *Language naturalness*. We evaluate 500 samples. Each evaluation unit is assigned to 3 workers to eliminate human variance. The first study attempts to evaluate how well the generated text correctly conveys information in the table, by counting the number of facts in the text supported by the table, and contradicting with or missing from the table. The 2nd and 3rd columns of Table 5 show the average number of supporting and contradicting facts for our method, comparing to the strongest baseline and the gold reference.

The second study evaluates whether the generated text is grammatically correct and fluent, regardless of factual correctness. We conduct pairwise comparison among all methods, and calculate the average times each method is chosen to be better than another, shown in the 4th column of Table 5. Our method brings a significant improvement over the strongest baseline ($p < 0.01$ in Tukey's HSD test for all measures). The copy loss term further alleviates producing incorrect facts. The language naturalness result of our method without the copy loss is slightly better, because this evaluation does not consider factual correctness; thus the generated texts with more wrong facts can still get high score. See Appendix C for more details of our evaluation procedure.

	# Supp.	# Cont.	Lan. Score
Gold Reference	4.25	0.84	1.85
Base + switch	2.57	2.17	0.93
Base + switch + LM (ours)	**3.64**	**1.12**	1.59
- w/o copy loss p_{copy}	3.54	1.30	**1.63**

Table 5: Human evaluation results: Average number of supporting facts (column 2, the larger the better), contradicting facts (column 3, the smaller the better), and language naturalness score (column 4, the larger the better).

5 Conclusion

In this paper, we propose the new research problem of few-shot natural language generation. Our approach is simple, easy to implement, while achieving strong performance on various domains. Our basic idea of acquiring language modeling prior can be potentially extended to a broader scope of generation tasks, based on various input structured data, such as knowledge graphs, SQL queries, etc. The deduction of manual data curation efforts for such tasks is of great potential and importance for many real-world applications.

Acknowledgment

We thank the anonymous reviewers for their thoughtful comments. We thank Shuming Ma for releasing the processed data and code for the Pivot model. This research was supported by the Intel AI Faculty Research Grant. The authors are solely responsible for the contents of the paper and the opinions expressed in this publication do not reflect those of the funding agencies.

References

Alison J Cawsey, Bonnie L Webber, and Ray B Jones. 1997. Natural language generation in health care.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bhuwan Dhingra, Manaal Faruqui, Ankur P. Parikh, Ming-Wei Chang, Dipanjan Das, and William W. Cohen. 2019. Handling divergent reference texts when evaluating table-to-text generation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4884–4895. Association for Computational Linguistics.

Chrysanne DiMarco, HDominic Covvey, D Cowan, V DiCiccio, E Hovy, J Lipa, D Mulholland, et al. 2007. The development of a natural language generation system for personalized e-health information. In *Medinfo 2007: Proceedings of the 12th World Congress on Health (Medical) Informatics; Building Sustainable Health Systems*, page 2339. IOS Press.

Hady ElSahar, Christophe Gravier, and Frédérique Laforest. 2018. Zero-shot question generation from knowledge graphs for unseen predicates and entity types. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 218–228.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The webnlg challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation, INLG 2017, Santiago de Compostela, Spain, September 4-7, 2017*, pages 124–133.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5110–5117.

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

He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. 2017. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1766–1776.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3294–3302.

Ioannis Konstas and Mirella Lapata. 2012. Unsupervised concept-to-text generation with hypergraphs. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 3-8, 2012, Montréal, Canada*, pages 752–761.

Ioannis Konstas and Mirella Lapata. 2013. A global model for concept-to-text generation. *J. Artif. Intell. Res.*, 48:305–346.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1188–1196.

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, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1203–1213.

Percy Liang, Michael I. Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*, pages 91–99.

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018. Table-to-text generation by structure-aware seq2seq learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and*

the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, pages 4881–4888.

Wei Lu, Hwee Tou Ng, and Wee Sun Lee. 2009. Natural language generation with tree conditional random fields. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 400–409.

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. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2047–2057. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119.

Jekaterina Novikova, Ondrej Dusek, 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, Saarbrücken, Germany, August 15-17, 2017*, pages 201–206.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2018. Data-to-text generation with content selection and planning. *CoRR*, abs/1809.00582.

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

Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87.

Amrita Saha, Vardaan Pahuja, Mitesh M. Khapra, Karthik Sankaranarayanan, and Sarath Chandar. 2018. Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 705–713.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Yu Su, Huan Sun, Brian M. Sadler, Mudhakar Srivatsa, Izzeddin Gur, Zenghui Yan, and Xifeng Yan. 2016. On generating characteristic-rich question sets for QA evaluation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 562–572.

Marilyn A. Walker, Owen Rambow, and Monica Rogati. 2001. Spot: A trainable sentence planner. In *Language Technologies 2001: The Second Meeting of the North American Chapter of the Association for Computational Linguistics, NAACL 2001, Pittsburgh, PA, USA, June 2-7, 2001*.

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2253–2263.

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2018. Learning neural templates for text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3174–3187.

Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, and Xiaoming Li. 2016. Neural generative question answering. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2972–2978.

Appendix A. Implementation Details

We use the Adam optimizer (Kingma and Ba, 2015) with learning rate set to 0.0003. The mini-batch size is set to 40 and the weight λ of the copy loss term to 0.7. The dimension of the position embedding is set to 5. For attribute name with multiple words, we average their word embeddings as the attribute name embedding. Refer to our released code and data at `https://github.com/czyssrs/Few-Shot-NLG` for more details.

Appendix B. ROUGE-4 Results

Following previous work (Liu et al., 2018), we conduct automatic evaluations using BLEU-4 and ROUGE-4 (F-measure)[3]. Table 6, 7 and 8 show the ROUGE-4 results for three domains *Humans*, *Books* and *Songs*, respectively.

Domain	Humans				
# of training instances	-	50	100	200	500
Template	5.1	-	-	-	-
Base-original	-	0.1	0.4	0.5	0.6
Base	-	0.1	0.4	0.8	1.5
Base+switch	-	4.9	6.3	9.8	12.5
Base+switch+LM-scratch	-	1.0	2.8	4.7	7.1
Base+switch+LM (Ours)	-	**14.1**	**16.2**	**22.1**	**28.3**

Table 6: ROUGE-4 results on *Humans* domain

Domain	Books				
# of training instances	-	50	100	200	500
Template	15.0	-	-	-	-
Base-original	-	1.1	1.6	2.1	1.5
Base	-	1.7	1.5	2.1	2.4
Base+switch	-	12.8	15.0	18.1	20.7
Base+switch+LM-scratch	-	2.4	4.2	6.5	10.7
Base+switch+LM (Ours)	-	**22.5**	**23.1**	**25.0**	**27.6**

Table 7: ROUGE-4 results on *Books* domain

Appendix C. Human Evaluation Details

We conduct human evaluation studies using Amazon Mechanical Turk, based on two aspects: *Factual correctness* and *Language naturalness*. For both studies, we evaluate the results trained with 200 training instances of *Humans* domain. We randomly sample 500 instances from the test set, together with the texts generated with different methods. Each evaluation unit is assigned to 3 workers to eliminate human variance.

[3]We use standard scripts NIST mteval-v13a.pl (for BLEU), and rouge-1.5.5 (for ROUGE)

Domain	Songs				
# of training instances	-	50	100	200	500
Template	24.5	-	-	-	-
Base-original	-	3.4	4.2	4.7	4.8
Base	-	4.1	5.1	4.7	5.8
Base+switch	-	20.2	21.7	23.2	24.8
Base+switch+LM-scratch	-	5.4	8.0	12.0	15.0
Base+switch+LM (Ours)	-	**26.2**	**28.6**	**30.1**	**32.6**

Table 8: ROUGE-4 results on *Songs* domain

The first study attempts to evaluate how well a generated text can correctly convey information in the table. Each worker is present with *both the input table and a generated text*, and asked to count how many facts in the generated text are supported by the table, and how many are contradicting with or missing from the table, similar as in (Wiseman et al., 2017). The we calculate the average number of supporting and contradicting facts for the texts generated by each method.

The second study aims to evaluate whether the generated text is grammatically correct and fluent in terms of language, regardless of factual correctness. Each worker is present with a pair of texts generated from the same input table, by two different methods, then asked to select the better one only according to language naturalness, or "Tied" if the two texts are of equal quality. *The input table is not shown to the workers*. Each time a generated text is chosen as the better one, we assign score of 1.0. If two texts are tied, we assign 0.5 for each. We then calculate the average score for the texts generated by each method, indicating its superiority in pairwise comparisons with all other methods.

The significance test is conducted respectively on all three measures: number of supporting facts and number of contradicting facts for the first study; the assigned score for the second study. We use the Tukey HSD post-hoc analysis of an ANOVA with the worker's response as the dependent variable, the method and worker id as independent variables.