

Few-Shot Table-to-Text Generation with Prototype Memory

Yixuan Su Zaiqiao Meng Simon Baker Nigel Collier

Language Technology Lab, University of Cambridge

{ys484, zm324, sb895, nhc30}@cam.ac.uk

Abstract

Neural table-to-text generation models have achieved remarkable progress on an array of tasks. However, due to the data-hungry nature of neural models, their performances strongly rely on large-scale training examples, limiting their applicability in real-world applications. To address this, we propose a new framework: Prototype-to-Generate (P2G), for table-to-text generation under the few-shot scenario. The proposed framework utilizes the retrieved prototypes, which are jointly selected by an IR system and a novel prototype selector to help the model bridging the structural gap between tables and texts. Experimental results on three benchmark datasets with three state-of-the-art models demonstrate that the proposed framework significantly improves the model performance across various evaluation metrics.

1 Introduction

Generating natural language from structured table (Gatt and Krahmer, 2018), i.e. table-to-text generation, is an important research problem for various NLP applications, such as biographical descriptions (Lebret et al., 2016), restaurant information (Novikova et al., 2017), basketball game summaries (Wiseman et al., 2017), and open-domain question answering (Chen et al., 2021).

The main challenge of table-to-text generation stems from the structural difference between the table and the natural language text. With recent advances in neural networks, many sophisticated neural models (Liu et al., 2018; Gehrmann et al., 2018; Puduppully et al., 2019a,b; Su et al., 2021b) have been proposed to address this problem. While achieving impressive results, such neural models are data-hungry, i.e. large amounts of training data are required for them to learn the mapping between tables and texts. This can prohibit these models from being applied to real-world applications due to the huge data curation overhead (Chen

et al., 2020b). This motivates us to investigate *few-shot table-to-text generation* (Ma et al., 2019; Chen et al., 2020b), that allows the model to learn a satisfactory table-to-text mapping with limited labelled training data.

In this work, we propose to address this problem by augmenting data-to-text generation models with prototype memory acquired from a large unlabelled corpus. Our motivation is two-fold: (1) Relevant human-authored texts, termed “prototypes”, are informative and can teach the model how to better describe the table when limited training data is available. (2) However, traditional lexical-based IR systems, e.g. BM25, are inaccurate and the quality of their results are not guaranteed. Therefore, a BERT-based prototype selector is required to further select the prototypes, from the results retrieved by the IR system, that are closely related to the table for better guiding the neural generation model.

Figure 1 illustrates the proposed Prototype-to-Generate (P2G) framework. Given the table, an IR system is first applied to retrieve candidates that are potentially related to the table from a large unlabelled corpus. Based on the retrieved candidates, a prototype selector then selects the top n prototypes based on the table-text pairwise similarity. Lastly, a sequence generator takes the table and the selected prototypes as input to produce the output. To prevent the model from uncritically copying the information contained in the prototypes that is irrelevant to the table, we introduce a content-aware learning objective when training the generator.

In recent years, retrieval-based (i.e. template-based) text generation has been studied in different NLP areas, including machine translation (Gu et al., 2017), unconditional text generation (Gua et al., 2018), dialogue systems (Wu et al., 2019; Su et al., 2021c), paraphrase generation (Kazemnejad et al., 2020; Su et al., 2021a), and question answering (Lewis et al., 2020b). Despite their differences, we identify two major limitations in previous studies

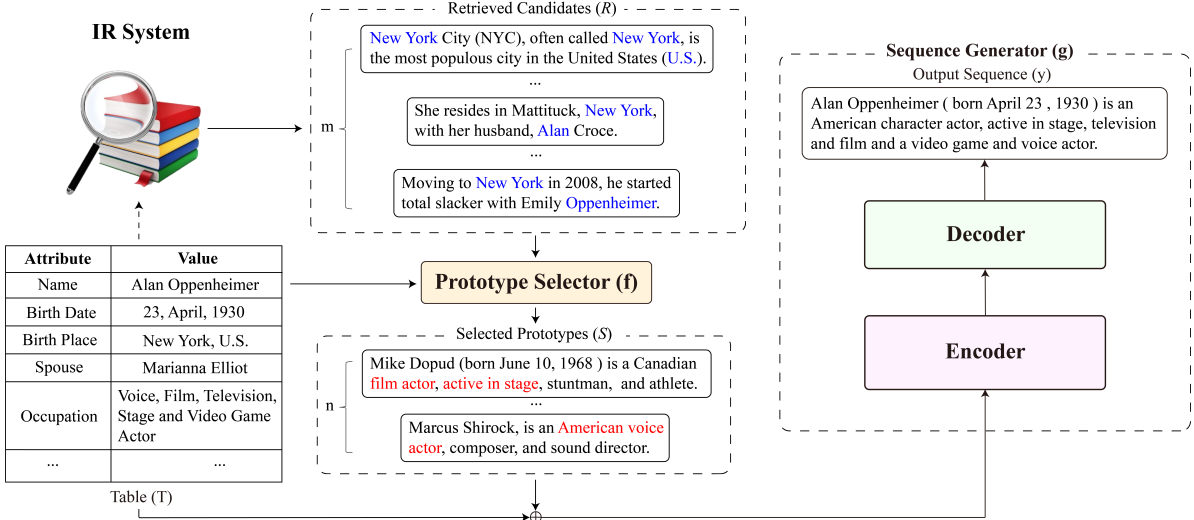


Figure 1: An overview of the proposed Prototype-to-Generate (P2G) framework.

compared to our approach. Firstly, most previous research (Gu et al., 2017; Wu et al., 2019; Kazemnejad et al., 2020) build their retrieval corpus based on data consisting of aligned source-target pairs, which precludes the use of abundant unlabelled data. Secondly, current retrieval mechanisms are either based on lexical similarity (e.g. BM25) where its accuracy cannot be guaranteed, or large neural networks (Karpukhin et al., 2020) which require a large amount of data to train.

Notably, our framework is independent of the choice of generation model. For a comprehensive evaluation, we test our approach on three representative models, including the current state of the art. The experimental results on three datasets show that our framework leads to remarkable performance improvements across all evaluation metrics.

2 Methodology

Figure 1 depicts an overview of our framework. Given a linearized table $T = \{t_1, \dots, t_{|T|}\}$, where $t_i = \{a_i, v_i\}$ is an attribute-value pair, an IR system first retrieves a set of m candidates \mathcal{R} from the large unlabelled corpus. Then, a prototype selector f (§2.1) selects the top n prototypes \mathcal{S} from \mathcal{R} that are most related to T . Lastly, a sequence generator g (§2.2) takes T and \mathcal{S} to produce the output y .

2.1 Prototype Selector

As illustrated in Figure 1, given the table T , the IR system relies on lexical features (e.g., word overlaps between the table and texts as colored in blue) to retrieve candidates \mathcal{R} . However, such lexical features are inaccurate and the semantic relevance

between T and \mathcal{R} cannot be guaranteed. To remedy this problem, we utilize a prototype selector f to select the top n prototypes \mathcal{S} from \mathcal{R} based on the table-text pairwise similarity. Formally, given the table T and a text $r \in \mathcal{R}$, their pairwise similarity score is defined as $f(T, r)$ and \mathcal{S} is then defined as:

$$\mathcal{S} = \arg \max_{\mathcal{R}' \in \mathcal{R}, |\mathcal{R}'|=n} \sum_{r \in \mathcal{R}'} f(T, r). \quad (1)$$

Figure 1 shows examples of the selected prototypes, \mathcal{S} . We see that \mathcal{S} are better related to the table and being closer to the reference text, i.e., the reference and \mathcal{S} could share similar contexts like the words in red. Thus, \mathcal{S} can be deemed as an guiding signal which teaches the model how to describe the table.

In this work, we use BERT (Devlin et al., 2019) to build the prototype selector. The score $f(T, r)$ is computed by a linear projection over the average embeddings of $\text{BERT}([T:r])$, where $[\cdot]$ denotes concatenation operation. During training, given the table T , the reference text y , and the retrieved candidate set \mathcal{R} provided by the IR system, the learning objective of the prototype selector is defined as:

$$\mathcal{L}_f = \sum_{j=1}^k \max\{0, 1 - f(T, y) + f(T, \mathcal{R}_j)\}, \quad (2)$$

where $\mathcal{R}_j \in \mathcal{R}$ and k is the number of negatives sampled from \mathcal{R} . After training f , we can obtain the prototype-augmented dataset $\mathcal{D} = \{(T, \mathcal{S}, y)_i\}_{i=1}^{|\mathcal{D}|}$ for the learning of the generator.

2.2 Sequence Generator

The proposed framework is model-agnostic, thus the generator g can be any generation model. Given

Domain	Humans				Books				Songs			
Training Size	50	100	200	500	50	100	200	500	50	100	200	500
<i>Retrieval-Based</i>												
Retri-Gen	7.4/0.7	10.3/1.6	13.2/2.7	16.5/4.1	12.1/1.8	13.2/2.0	14.7/2.4	15.9/3.3	13.4/2.7	14.3/3.1	16.2/4.3	17.7/4.9
RA-Gen	29.4/15.8	33.6/18.9	40.1/26.7	44.3/30.9	34.7/22.2	35.7/22.9	37.4/24.9	40.9/28.3	34.9/24.8	36.4/26.1	39.0/29.2	42.1/31.7
Struct-Aware [‡]	2.9/0.1	5.1/0.4	6.1/0.8	8.3/1.5	7.3/1.7	6.8/1.5	7.8/2.1	8.8/2.4	10.4/4.1	12.0/5.1	11.6/4.7	13.1/5.8
Pivot	14.9/3.2	18.7/6.9	25.3/14.1	29.8/17.3	23.1/10.7	24.9/13.3	27.0/15.2	29.8/18.1	26.2/14.7	28.0/16.2	29.2/17.7	31.7/20.0
KGPT	30.2/18.8	35.0/22.8	38.9/26.1	43.7/30.4	35.3/24.2	37.4/25.8	38.4/26.7	42.0/29.2	37.9/28.3	39.8/30.1	40.3/30.5	42.9/33.0
Switch-GPT [†]	25.7/14.1	29.5/16.2	36.1/22.1	41.7/28.3	34.3/22.5	36.2/23.1	37.9/25.0	40.3/27.6	36.1/26.2	37.2/28.6	39.4/30.1	42.2/32.6
Table-GPT [‡]	29.8/16.3	34.5/20.6	40.6/27.6	45.6/32.4	35.1/24.0	37.3/25.4	38.5/26.7	41.6/28.9	36.7/27.1	37.8/29.4	39.3/30.6	42.3/32.8
T5-Prefix	32.6/20.7	37.1/23.1	41.7/28.8	46.3/33.2	34.2/21.2	38.3/26.7	39.4/27.6	42.9/30.0	37.6/28.1	38.7/29.2	40.0/30.3	43.5/33.9
P2G+Switch-GPT	31.4/19.9	36.5/22.7	42.0/30.1	45.8/32.6	38.2/25.4	39.9/27.3	41.7/29.2	44.6/31.7	39.1/29.9	40.3/30.7	41.8/32.0	45.0/35.4
P2G+Table-GPT	34.9/23.2	38.9/25.1	43.1/31.2	48.1/35.0	40.1/29.3	41.0/28.6	43.1/30.4	47.0/34.0	41.2/31.7	42.7/33.6	44.2/34.9	47.9/38.1
P2G+T5-Prefix	39.3/27.9	42.6/30.8	46.2/34.0	50.1/37.3	41.2/28.3	43.4/30.5	46.4/33.8	49.2/36.1	42.8/33.0	45.9/35.7	47.6/37.5	50.7/40.1

Table 1: Results on datasets from three domains. In each entry, x/y denotes the model performance on BLEU-4/ROUGE-4(F-measure). [†] and [‡] results are copied from Chen et al. (2020b) and Gong et al. (2020). All results acquired with the proposed framework outperform the original model with a significance level p -value < 0.01 .

a training example $(T, \mathcal{S}, y) \in \mathcal{D}$, the learning of g is defined as: $\mathcal{L}_{\text{LM}} = -\sum_{i=1}^{|y|} \log p_{\theta}(y_i|y_{<i}; X)$, where θ denotes the parameters of the generator, and $X = [T:\mathcal{S}]$. Moreover, we introduce a new content-aware learning objective. Our motivation is that the prototypes \mathcal{S} is likely to contain information that is irrelevant to the table, thus the generator should learn to ignore the irrelevant part of \mathcal{S} and only focus on the useful information. To this end, inspired by Welleck et al. (2020), we formulate the content-aware learning objective as: $\mathcal{L}_{\text{CA}} = -\sum_{i=1}^{|y|} \sum_{\tilde{y} \in \mathcal{S}, \tilde{y} \neq y} \log(1 - p_{\theta}(\tilde{y}|y_{<i}; X))$ which discourages the generation of the irrelevant tokens contained in \mathcal{S} . The generator overall learning objective is then defined as: $\mathcal{L}_g = \mathcal{L}_{\text{LM}} + \mathcal{L}_{\text{CA}}$.

3 Experiment

3.1 Experiment Setup

We conduct experiments on three benchmark few-shot table-to-text datasets (Chen et al., 2020b) from different domains: *Humans*, *Books*, and *Songs*. Following previous studies (Chen et al., 2020b; Gong et al., 2020), we train our model on different settings by varying the training size from $\{50, 100, 200, 500\}$, and evaluate our model using BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) metrics. Test sets of *Humans*, *Books*, and *Songs* contain 13587, 5252 and 11879 instances.

To build the IR system, we use Lucene¹ to pre-index all sentences contained in the English Wikipedia (Dec. 2018 dump). For each table, the IR system retrieves 100 sentences as the candidates \mathcal{R} . The prototype selector then select the top 3 re-

sults from \mathcal{R} as the prototypes \mathcal{S}^2 . When training the prototype selector, we set k in Eq. (2) as 5.

We compare our approach with both existing table-to-text methods that are not retrieval-based and also with the existing retrieval-based methods which we adapt for our concerned task. The existing table-to-text methods include Struct-Aware (Liu et al., 2018), Pivot (Ma et al., 2019), Switch-GPT (Chen et al., 2020b), KGPT (Chen et al., 2020a), Table-GPT (Gong et al., 2020), and T5-Prefix (Ribeiro et al., 2020). The latter four are based on pre-trained language models (PLMs). The retrieval-based approaches include Retri-Gen (Wu et al., 2019) and RA-Gen (Lewis et al., 2020b), where RA-Gen is based on PLMs. We select three representative models (Switch-GPT, Table-GPT, and T5-Prefix) to test the proposed framework.

3.2 Main Results

Table 1 lists the experiment results, where P2G+X indicates using model X under our framework. We can see that the proposed framework consistently and significantly improves the performance of all three models on all metrics, showing the robustness and universality of our approach. The notable performance gains suggest that the incorporation of retrieved prototypes greatly benefit the model’s ability in bridging the gap between tables and texts. It is worth noting that the RA-Gen model applies a strong BART (Lewis et al., 2020a) as the generator. However, their retrieval module is purely based on a large neural models (Karpukhin et al., 2020) that requires a large amount of data to train, and its accuracy degenerates when training data is limited,

²To avoid the data leakage problem, when building the dataset, we make sure the prototypes do not contain the reference.

¹<https://lucene.apache.org/core/>

Training Size	50	100	200	500
T5-Prefix	32.6/20.7	37.1/23.1	41.7/28.8	46.3/33.2
+Ret	32.9/21.2	37.4/23.5	42.1/29.0	46.7/33.4
+Ret&PS	38.8/27.0	42.0/30.2	45.8/33.5	49.2/36.6
+Ret&PS&CA	39.3/27.9	42.6/30.8	46.2/34.0	50.1/37.3

Table 2: Ablation study results on *Humans* dataset. In each entry, x/y denotes the BLEU-4/ROUGE-4 scores.

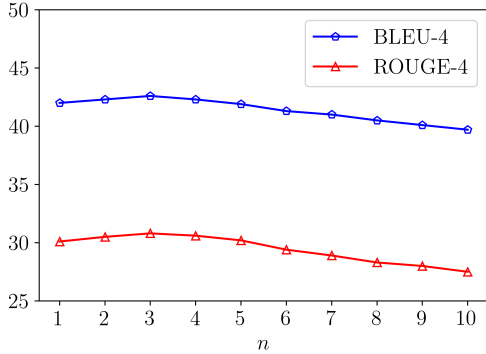


Figure 2: Effect of the number of prototypes (n).

leading to the reduced generation performance.

3.3 Further Analysis

In this section, we present further discussions and empirical analysis of the proposed model.

Ablation Study. First, we perform ablation analysis on the T5-Prefix model by progressively incorporating each proposed technique. The +Ret model directly utilizes the top 3 retrieved results from the IR system as input. The +Ret&PS model utilizes the prototypes selected by the prototype selector as input. Finally, we include the proposed content-aware objective (+Ret&PS&CA) which results in the same model as P2G+T5-Prefix. The experiments are conducted on the *Humans* dataset with different training size. Table 2 lists the results which show that each component positively contributes to the overall performance. By comparing T5-Prefix with +Ret, we only observe a marginal improvement, suggesting that the retrieved results from the IR system are inaccurate (i.e., unrelated to the table) which brings little help to the generator. Next, from the results of +Ret&PS model we see that the incorporation of prototype selector significantly boosts the performance. This is inline with our hypothesis that the prototype selector can select more accurate (i.e., related to the table and similar to the reference) prototypes that can effectively teach the generator about how to describe the table. Lastly, the results of +Ret&PS&CA show that the proposed content-aware learning objective

	#Support \uparrow	#Contradict \downarrow	Fluency \uparrow
Agreement	0.64	0.61	0.53
Reference	4.27	0.31	1.85
Switch-GPT	3.23	0.98	1.37
Table-GPT	3.47	0.75	1.42
T5-Prefix	3.59	0.62	1.58
P2G+T5-Prefix	3.98	0.47	1.71

Table 3: Human Evaluation Results. \uparrow means the higher the better and \downarrow means the lower the better.

also benefits the model performance.

Effect of the Number of Prototypes. Next, we examine how the number of prototypes (n in Eq. (1)) affects the model performance. To this end, we train P2G+T5-Prefix with 100 instances on the *Humans* dataset by varying the size of n . Figure 2 depicts the results of BLEU and ROUGE. We observe that, when n is small (i.e., $n \leq 3$), the model performances are relatively the same. However, as n approaching 10, the results drop notably. The reason is that, as n increases, the top n prototypes are likely to contain more information that is irrelevant to the table (i.e. noisy information), which leads to the degeneration of model performances.

3.4 Human Evaluation

We also conduct a human evaluation to assess the P2G+T5-Prefix model against several strong baselines, using graders proficient in English from an internal grading platform. Experiments are conducted on *Humans* dataset using 100 training instances and we randomly select 300 test cases for evaluation. All generated results, plus the reference, are evaluated by three graders on two aspects: (1) *factual correctness*; and (2) *language fluency*. Firstly, the graders are asked to count how many facts contained in the output are consistent with the table (#Support), and are contradicted to the table (#Contradict). Secondly, the graders are asked to assess the output in terms of language fluency on a 3-point Likert scale (0, 1, or 2).

Table 3 lists the evaluation results, with the first row showing strong inter-annotator agreements as measured by Fleiss’ kappa coefficient (Fleiss et al., 1971). The results show that our model (P2G+T5-Prefix) significantly outperforms other baseline models on all metrics (Sign Test with p-value < 0.05). The performance gains of P2G+T5-Prefix over T5-Prefix further suggest that the prototypes help the model to produce not only more syntactically fluent but also more factually correct outputs.

Table:	Name [The Absence] Background [Grouporband] Origin [Tampa, Florida, U.S.] Genre [melodic death metal, thrash metal] Years Active [2002–present] Current Members [Jamie Stewart, Patrick Pintavalle, Mike Leon, Jeramie Kling, Per Nilsson] Past Members [Justin Reynolds, Nicholas Calaci, John Allen, Chris Pistillo, Peter Joseph]
Reference:	The Absence is an American melodic death metal band from Tampa, Florida.
T5+Prefix:	Jamie Stewart Patrick Pintavalle Mike Leon Jeramie Kling Per Nilsson, current members is Justin Reynolds Nicholas Calaci John Allen Chris Pistillo Peter Joseph.
P2G+T5-Prefix	
Prototypes:	1: One Man Army and the Undead Quartet was a Swedish band , that played a fusion of melodic death metal and thrash metal. 2: Epoch of Unlight is a melodic death metal band from Memphis, Tennessee. 3: Inactive Messiah is a Greek melodic death metal band , founded in Athens.
Output:	The Absence is an American melodic death metal band from Tampa, Florida, U.S.
Table:	Name [Axel Toupane] Position [shooting guard/small forward] Height ft [6] Height in [7] Weight lb [197] League [NBA] Team [Toronto Raptors] Nationality [French] Draft Year [2014] Birth Date [23 July 1992] Birth Place [Mulhouse, France] Career Start [2011] Years [2011–2015]
Reference:	Axel Toupane (born July 23, 1992) is a French professional basketball player who currently plays for the Toronto Raptors of the National Basketball Association (NBA).
T5+Prefix:	Axel Toupane (born 23 July 1992) is a French professional basketball player.
P2G+T5-Prefix	
Prototypes:	1: Shannon Scott (born December 21, 1992) is an American professional basketball player who currently plays for the Toronto Raptors . 2: Bismack Biyombo Sumba (born August 28, 1992) is a Congolese professional basketball player who currently plays for the Toronto Raptors of the National Basketball Association . 3: Jama Mahlalela (born in Swaziland) is an assistant coach for the Toronto Raptors of the NBA .
Output:	Axel Toupane (born July 23, 1992) is a French professional basketball player in the team of the Toronto Raptors of the National Basketball Association (NBA) .

Table 4: Examples of generated result from *Humans* dataset. (best viewed in color)

4 Case Study

In Table 4, we present two generated examples from our model. For comparison, we also show the results generated by the strongest baseline (T5-Prefix) along with the reference sentence. As for our model, we show the selected prototypes along with the generated output. Both our model and the baseline model are trained with 100 instances.

As seen in the first case, the T5-Prefix fails to produce a correct output which describes the band. Instead, it just elaborates the name of the band members based on the table. In contrast, by relying on the prototypes that are related to the table, our model (P2G+T5-Prefix) produces an output that properly describes the band. Similarly, in the second case, the result of our model is more diverse and contains more facts that are supported by the table. These results further demonstrate that the prototypes can be deemed as effective guiding signals which teach the model how to describe the table. For better illustration, we highlight the parts, with red color, of prototypes on which the model relies when producing the output.

5 Conclusion

In this study, we introduced a new retrieval-based framework, Prototype-to-Generate (P2G),

which augments table-to-text models with prototype memory from unlabelled data. Extensive experiments and analysis on three benchmark datasets show that our approach can significantly improve the performance of various strong generation models on all evaluation metrics. Our code, models and other related resources can be found in <https://github.com/yxuansu/Few-Shot-Table-to-Text-Generation>

Acknowledgments

The authors wish to thank our anonymous reviewers for their suggestions and comments.

References

- Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Yang Wang, and William W. Cohen. 2021. [Open question answering over tables and text](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. 2020a. [KGPT: knowledge-grounded pre-training for data-to-text generation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 8635–8648. Association for Computational Linguistics.

- Zhiyu Chen, Harini Eavani, Wenhua Chen, Yinyin Liu, and William Yang Wang. 2020b. [Few-shot NLG with pre-trained language model](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 183–190. Association for Computational Linguistics.
- Émilie Colin and Claire Gardent. 2019. [Generating text from anonymised structures](#). In *Proceedings of the 12th International Conference on Natural Language Generation, INLG 2019, Tokyo, Japan, October 29 - November 1, 2019*, pages 112–117. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.
- J.L. Fleiss et al. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.
- Albert Gatt and Emiel Krahmer. 2018. [Survey of the state of the art in natural language generation: Core tasks, applications and evaluation](#). *J. Artif. Intell. Res.*, 61:65–170.
- Sebastian Gehrmann, Falcon Z. Dai, Henry Elder, and Alexander M. Rush. 2018. [End-to-end content and plan selection for data-to-text generation](#). In *Proceedings of the 11th International Conference on Natural Language Generation, Tilburg University, The Netherlands, November 5-8, 2018*, pages 46–56. Association for Computational Linguistics.
- Heng Gong, Yawei Sun, Xiaocheng Feng, Bing Qin, Wei Bi, Xiaojiang Liu, and Ting Liu. 2020. [Tablept: Few-shot table-to-text generation with table structure reconstruction and content matching](#). In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 1978–1988. International Committee on Computational Linguistics.
- Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O. K. Li. 2017. [Search engine guided non-parametric neural machine translation](#). *CoRR*, abs/1705.07267.
- Kelvin Guu, Tatsunori B. Hashimoto, Yonatan Oren, and Percy Liang. 2018. [Generating sentences by editing prototypes](#). *Trans. Assoc. Comput. Linguistics*, 6:437–450.
- Mihir Kale. 2020. [Text-to-text pre-training for data-to-text tasks](#). *CoRR*, abs/2005.10433.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. [Dense passage retrieval for open-domain question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.
- Amirhossein Kazemnejad, Mohammadreza Salehi, and Mahdieh Soleymani Baghshah. 2020. [Paraphrase generation by learning how to edit from samples](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6010–6021. Association for Computational Linguistics.
- Ravi Kondadadi, Blake Howald, and Frank Schilder. 2013. [A statistical NLG framework for aggregated planning and realization](#). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 1406–1415. The Association for Computer Linguistics.
- 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. The Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. [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, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.
- Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. [Retrieval-augmented generation for knowledge-intensive NLP tasks](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- 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. AAAI Press.
- 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.
- 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. Association for Computational Linguistics.
- Alice H. Oh and Alexander I. Rudnicky. 2000. [Stochastic language generation for spoken dialogue systems](#). In *ANLP-NAACL 2000 Workshop: Conversational Systems*.
- 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, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.
- Ratish Puduppully, Li Dong, and Mirella Lapata. 2019a. [Data-to-text generation with content selection and planning](#). In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6908–6915. AAAI Press.
- Ratish Puduppully, Li Dong, and Mirella Lapata. 2019b. [Data-to-text generation with entity modeling](#). 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 2023–2035. Association for Computational Linguistics.
- Ehud Reiter and Robert Dale. 1997. [Building applied natural language generation systems](#). *Nat. Lang. Eng.*, 3(1):57–87.
- Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2020. [Investigating pretrained language models for graph-to-text generation](#). *CoRR*, abs/2007.08426.
- Amanda Stent, Rashmi Prasad, and Marilyn Walker. 2004. [Trainable sentence planning for complex information presentations in spoken dialog systems](#). In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 79–86, Barcelona, Spain.
- Yixuan Su, David Vandyke, Simon Baker, Yan Wang, and Nigel Collier. 2021a. [Keep the primary, rewrite the secondary: A two-stage approach for paraphrase generation](#). In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, pages 560–569. Association for Computational Linguistics.
- Yixuan Su, David Vandyke, Sihui Wang, Yimai Fang, and Nigel Collier. 2021b. [Plan-then-generate: Controlled data-to-text generation via planning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics.
- Yixuan Su, Yan Wang, Deng Cai, Simon Baker, Anna Korhonen, and Nigel Collier. 2021c. [PROTOTYPE-TO-STYLE: dialogue generation with style-aware editing on retrieval memory](#). *IEEE ACM Trans. Audio Speech Lang. Process.*, 29:2152–2161.
- Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. [Neural text generation with unlikelihood training](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.
- 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. Association for Computational Linguistics.
- 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. Association for Computational Linguistics.
- Yu Wu, Furu Wei, Shaohan Huang, Yunli Wang, Zhoujun Li, and Ming Zhou. 2019. [Response generation by context-aware prototype editing](#). In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 7281–7288. AAAI Press.
- Rong Ye, Wenxian Shi, Hao Zhou, Zhongyu Wei, and Lei Li. 2020. [Variational template machine for data-to-text generation](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

A Related Work

Table-to-Text Generation. Table-to-text generation is a long-standing problem (Reiter and Dale, 1997) that aims at producing natural language descriptions of structured table. Traditional systems are primarily built on template-based algorithms (Oh and Rudnicky, 2000; Stent et al., 2004; Konradadi et al., 2013). With recent advances in neural networks, researchers have built different neural models based on various strategies, e.g. latent-variables (Wiseman et al., 2018; Ye et al., 2020), structure awareness (Liu et al., 2018; Colin and Gardent, 2019), copy mechanism (Gehrmann et al., 2018; Puduppully et al., 2019a,b), and pre-trained language models (PLMs) (Chen et al., 2020a; Kale, 2020; Ribeiro et al., 2020). More recently, to alleviate the data-hungry nature of neural models, Ma et al. (2019) applied a pipeline model which first selects key facts from the table before producing the output. Chen et al. (2020a) designed a knowledge-grounded strategy for language model pre-training. Chen et al. (2020b) and Gong et al. (2020) adapted the pre-trained GPT-2 model with different architectural designs, e.g. switch policy (Chen et al., 2020b) and content matching (Gong et al., 2020), to address the few-shot table-to-text generation problem.

Retrieval-Based Text Generation. In the last few years, retrieval-based text generation has attracted much attention. Gu et al. (2017) utilized a search engine to assist the neural machine translation model. Guu et al. (2018) addressed unconditional text generation with a neural editor model that edits the retrieved prototypes. Wu et al. (2019) and Su et al. (2021c) incorporated retrieval frameworks into Seq2seq models to enrich the information contained in the dialogue responses. Kazemnejad et al. (2020) applied a retrieval model to assist the generation of paraphrased sentence. Lewis et al. (2020b) incorporated external knowledge using a retrieval model for knowledge-intensive question answering. To the best of our knowledge, our work is the first one which explores how retrieval-based approach could benefit neural models for table-to-text generation task.

B Human Evaluation Guidelines

In the human evaluation, the graders are asked to assess the results from two aspects. Following previous research (Chen et al., 2020b; Gong et al., 2020),

in the first study, the graders evaluate the *factual correctness* of the generated results by counting how many facts contained in the output are consistent with the table (#Support), and are contradicted to the table (#Contradict). In the second study, the graders assess the *language fluency* of the generated results following a 3-point Likert scale (0, 1, or 2). The definitions of different scores are provided as following:

- 2: The result is grammatically fluent and is easy to understand.
- 1: The result contains small errors but the errors does not affect your understanding.
- 0: The result does not make sense and it is unreadable.