Structure-Aware Pre-Training for Table-to-Text Generation

Xinyu Xing and Xiaojun Wan
Wangxuan Institute of Computer Technology, Peking University
The MOE Key Laboratory of Computational Linguistics, Peking University
{xingxinyu,wanxiaojun}@pku.edu.cn

Abstract

Table-to-text generation is a subtask of data-to-text generation which aims to generate naltural language text based on input table. Pre-training techniques have achieved great success on table-to-text generation. However, the pre-trained models used in previous works are typically trained on free-form natural language text while the input of table-to-text task is structured table. In this paper, we propose STTP, a pre-trained model that is trained with tables and their contexts. The STTP model can understand the structured input table and generate fluent text. Experiments on two datasets show the efficacy of our model.

1 Introduction

Data-to-text generation (Reiter and Dale, 1997) is an important natural language generation task with many practical applications, and it refers to the task of generating textual output from non-linguistic input data. The input data of the task can include tables of records, simulations of physical systems, spreadsheets, and so on. The output of the task is a natural language text. Datasets in common use include WEATHERGOV(Liang et al., 2009), ROTOWIRE(Wiseman et al., 2017), WebNLG(Gardent et al., 2017) and so on. Neural generation models with different improvements have achieved impressive results on data-to-text task. Table-to-text generation is a subtask of data-to-text generation which takes tables as input.

The pretrain-and-finetune framework, which refers to first pre-training a high capacity model on large corpora and then fine-tuning it on a downstream task, has outperformed prior state of the art on both natural language understanding task and natural language generation task. Inspired by the success of transfer learning, recently some works (Mager et al., 2020; Kale, 2020; Ribeiro et al., 2020) try to apply the pretrain-and-finetune framework on data-to-text generation. They fine-tuned the pre-trained model such as BART(Lewis et al., 2019) or T5(Raffel et al., 2019) on several downstream data-to-text tasks and achieved state-of-the-art results.

Although transfer learning has achieved great success on data-to-text generation, the pre-trained models used in previous works are typically trained on free-form natural language texts while the input of table-to-text task is structured table. The text-to-text pre-trained models learn a lot of knowledge and good language models from large amount of texts, so they work well on data-to-text generation task. But they still lack the ability to understand the structured data. So we propose a structure-aware table-to-text pre-trained model STTP which is trained with tables and their contexts for table-to-text task. STTP is built on top of the text-to-text pre-trained model BART, and it can understand the structured table and describe it with natural language text. We train our model based on BART because we hope our model can benefit from the knowledge and language model learned from large corpora. We propose three self-supervised tasks to train our model with large amount of tables and their contexts. The first self-supervised task is masked table language model (MTLM) which is like the classic MLM of BERT. The second self-supervised task is adjacent cell prediction (ACP) which refers to predict the cells around the current cell. The third self-supervised task is context reconstruction (CR) which refers to reconstructing the context of a table given the table and its broken context. The first two tasks aim to train the model to better understand the structured table, while the latter task aims to align the table and text. We use the tables extracted from WDCWebTable Corpus(Lehmberg et al., 2016) and their contexts to train our model. Experimental results on WEATHERGOV dataset and WebNLG dataset show the

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 2273–2278
August 1–6, 2021. ©2021 Association for Computational Linguistics

efficacy of our model.

The main contributions of this work are:

- We propose a structure-aware table-to-text pre-trained model STTP which is trained with three self-supervised tasks.

- Experimental results on WEATHERGOV dataset and WebNLG dataset show the efficacy of our model. Code will be released at `https://github.com/XingXinyu96/STTP`.

2 Related work

Data-to-text generation task involves taking structured data as input and generating text that describes this data. Traditional approaches (Stent et al., 2004; Walker et al., 2007) deal with the task in two steps: the selection of a subset of the input data to discuss and the surface realization of a generation. More recent works combine both steps by learning content plan and surface realization jointly with end-to-end models (Wen et al., 2015; Peng et al., 2017). Although the end-to-end model has achieved good results, many models (Perez-Beltrachini and Lapata, 2018; Sha et al., 2018; Puduppully et al., 2019) consider adding content selection and content planning modules to the end-to-end framework to improve performance. A lot of other new models with different improvements (Wiseman et al., 2018; Li and Wan, 2018; Liu et al., 2018; Roberti et al., 2019; Rebuffel et al., 2020) are proposed to explore how to build an effective data-to-text generator.

Inspired by the success of the pre-trained models in other natural language generation tasks, Harkous et al. (2020), Kale (2020) and Ribeiro et al. (2020) achieve state-of-the-art results on different data-to-text benchmarks with different pre-trained models. However, the existing pre-trained models are usually designed to generate text based on text input, thus lacking the ability to understand structured inputs. Several pre-training methods designed for table-to-text task have been proposed. Deng et al. (2020) present a weakly supervised Structure-Grounded pretraining framework (STRUG) for text-to-SQL that can effectively learn to capture text-table alignment. But their model is only for text-to-SQL task and need parallel text-table data. Yin et al. (2020) propose TABERT, a pretrained model which is trained with large amount of tables with their context. Their model is also used for text-to-SQL task. Chen et al. (2020a) propose a knowledge-grounded pre-trained (KGPT) model which is trained on a massive knowledge- grounded text corpus crawled from the web. Li et al. (2020) propose two self-supervised tasks, Number Ordering and Significance Ordering, to help to learn better table representation.

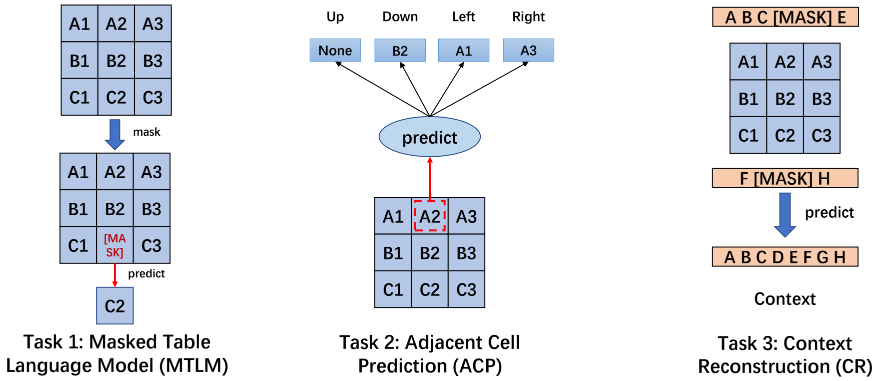
3 Approach

We use the same model architecture as BART, and add several classification layers on top of the encoder for our new self-supervised tasks. We train our model based on the text-to-text pre-trained model BART instead of training from scratch because we hope our model can benefit from the knowledge and language model BART learned from large corpus of text. The three self-supervised tasks are shown in Figure 1.

3.1 Self-Supervised Tasks

Task 1: Masked Table Language Model (MTLM). Inspired by the classic Masked Language Model proposed by BERT, we propose the Masked Table Language Model (MTLM) to learn the representation of input table. During pre-training, we treat the table as a sequence and randomly replace 15% of tokens in the table with [MASK] symbols and then the final hidden vectors corresponding to the mask tokens are fed into an output softmax over the vocabulary to predict the original tokens. In this way, our STTP model learns to understand the structured table. Although we do not explicitly consider the structure of the input table in this task, it is obvious that our model needs to understand the structure of the table to predict the masked tokens.

Task 2: Adjacent Cell Prediction (ACP). The previous MTLM task does not explicitly consider the structure of the input table, so we propose a new task Adjacent Cell Prediction (ACP) to explicitly help our model better understand the structure of the table. For a cell in a table, the surrounding cells are usually very important to understand it. So we feed the hidden vectors of a cell into several output layers to predict its top, bottom, left and right cells. In other words, we hope that the hidden vector of each cell can contain the information of other cells in the same row or column. This task requires our model to focus more on the relationship between cells in the same row or column when encoding the table. For efficiency, we only use the nearest cells in the same row or column of each cell as targets



Task 1: Masked Table Language Model (MTLM)

Task 2: Adjacent Cell Prediction (ACP)

Task 3: Context Reconstruction (CR)

Figure 1: Three Self-supervised Tasks for our STTP Model. Table is shown in blue color and context text is shown in wheat color.

for prediction.

Task 3: Context Reconstruction (CR). The previous two tasks focus on better understanding the structured table, but for table-to-text generation tasks, another important aspect is the alignment of table and text. Through the previous two tasks, our model can get a better representation of the table, then we need to generate text based on this representation which needs the alignment of table and text. If we only train the encoder with the previous two tasks and do not consider the alignment of table and text, we might get a better representation of input table but the representation is hard to understand for the well-trained decoder provided by BART. Table-text alignment data are difficult to obtain, so we use the table with text context (which is usually not strictly aligned with the table, but somewhat relevant with the table) to help our model align table and text. We randomly mask 15% tokens of the context and reconstruct the broken context with our model. There also exists a mismatch between pre-training and fine-tuning, since the input of our model is usually just a table without its context during fine-tuning. Therefore, we also train our model with a task to predict context based only on the input table. Since the table and its context are not exactly aligned, it is difficult to predict the context only from the table, but this task can help our model mitigate the mismatch problem.

3.2 Dataset for Pre-training

The first two tasks only need unsupervised tables, while the third task needs tables with context text. Yin et al. (2020) collect tables and their surrounding text from English Wikipedia and the WDCWebTable Corpus(Lehmberg et al., 2016) to train their model. The data they use is also suitable for our task. We use the preprocessing tool they provided to handle the WDCWebTable Corpus and get a lot of tables with context. Then we filter out the data with low matching degree between table and context, because such data are difficult to train for the third task. In addition, we filter out tables that contains too many numbers. Finally, we get 800k tables with context. We linearize the structure of tables to be compatible with our model. We add a special token [cell] in the middle of cells in the same row and add another special token [row] in the middle of rows of the table. When training task 3, the input of model includes both text and table, so we concatenate them together and add a special token [TABLE] between text and table.

3.3 Pre-training Procedure

We alternately train our model with the previously mentioned three self-supervised tasks. The first two tasks are only used to train the encoder while keeping the decoder frozen. The third task is divided into two subtasks in practice: one is to reconstruct the context given the table with its damaged context; the other one is to predict the context given only the table. Both of the two subtasks train the encoder and decoder at the same time. Considering that the latter subtask is very difficult because the context of a table is difficult to be predicted in many cases, we reduce the times of training it.

4 Experiment

4.1 Dataset

We perform the experiments on WEATHERGOV dataset(Liang et al., 2009) and WebNLG dataset(Gardent et al., 2017). In the WEATHERGOV dataset, the output text is a weather report, and the source data provides a structured representation of the temperature, sky conditions, etc. The WEATHERGOV dataset consists of 29, 528 scenarios, each with 36 weather records paired with a natural language weather forecast (28.7 avg. word length). The WebNLG challenge consists of mapping sets of RDF triples to text. The newest WebNLG dataset contains 16, 095 data inputs and 42, 873 data-text pairs. The average length of the output text is 22.3 words. We convert the input data into a table and then linearize the structure of table like what we do when pre-training.

4.2 Results

4.2.1 Results on WeatherGov Dataset

We use a batch size of 4 and finetune for 100 epochs over the WeatherGov Dataset. Results are presented in Table 1. As is shown in this table, seq2seq model(Mei et al., 2015) has achieved very good results, but the pre-trained model further improves the results greatly. The BLEU scores of the pre-trained models are more than 80, which indicates that the generated text is highly similar to the gold text. BART-Retrain refers to finetuning BART on both our pre-training dataset and the datasets in downstream tasks. Our STTP model outperforms the BART-Retrain model, which proves the improvements of STTP model over BART model is from the proposed training objective instead of the additional training data.

Model	BLEU	METEOR
(Mei et al., 2015)	61.01	n/a
BART-base	81.54	54.81
BART-Retrain	81.63	55.50
STTP	**82.63**	**56.35**

Table 1: Results on WeatherGov Dataset.

4.2.2 Results on WebNLG Dataset

We use a batch size of 4 and fine-tune for 16 epochs over the WebNLG Dataset. Results are presented in Table 2. The results of Seq2Seq, Seq2Seq+Delex and Seq2Seq+copy are copied from (Shimorina and Gardent, 2018). The results of GCN and KGPT-Seq are copied from (Chen et al., 2020b). As is shown in this table, all pre-trained models outperform the models without pre-training even if some of them do not explicitly consider the structure of the input data. This is due to the pre-trained models learn a lot of external knowledge and a good language model from large corpora. The structure-aware model like GCN outperforms the normal seq2seq model, which shows that structure understanding is important in this task. Our model further outperforms the BART-base model and the KGPT model, which show the efficacy of our model with new self-supervised tasks.

Model	BLEU	METEOR
Seq2Seq	54.0	37.0
Seq2Seq+Delex	56.0	39.0
Seq2Seq+Copy	61.0	42.0
GCN	60.80	42.76
KGPT-Seq	64.11	46.30
BART-BASE	62.62	43.28
BART-Retrain	62.89	43.34
STTP	**64.92**	**46.48**

Table 2: Results on WebNLG Dataset

We randomly sample 50 instances from the WebNLG dataset and perform human evaluation on them. Three graduate students are employed to rank the generated texts produced by each model in three aspects: readability (whether the generated text is fluent), accuracy (whether the information of the generated texts is consistent with that contained in the input table) and overall quality. We use Best-Worst Scaling (Louviere et al., 2015), which has been shown to produce more reliable results than ranking scales (Kiritchenko and Mohammad, 2017). Specifically, each score is computed as the percentage of times it was selected as best minus the percentage of times it was selected as worst, and ranges from -1 (unanimously worst) to +1 (unanimously best). Human evaluation results on WebNLG dataset are shown in Table 3. We can see our model outperforms KGPT model and BART-base model, which further demonstrates the efficacy of our method. Running examples are provided in the supplementary materials.

Model	Readability	Accuracy	Overall
KGPT-Seq	0.09	0.01	0.04
BART-base	-0.22	-0.10	-0.17
STTP	**0.13**	**0.09**	**0.13**

Table 3: Human Evaluation Results on WebNLG Dataset

5 Conclusion

In this paper, we propose STTP, a pre-trained model trained with tables and their contexts. STTP model has achieved great performance on two downstream tasks. In the future work, we hope to collect more data and try other self-supervised tasks to train more effective model for table-to-text task.

Acknowledgments

This work was supported by National Natural Science Foundation of China (61772036), Beijing Academy of Artificial Intelligence (BAAI) and Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). Xiaojun Wan is the corresponding author.

References

Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. 2020a. Kgpt: Knowledge-grounded pre-training for data-to-text generation. *arXiv preprint arXiv:2010.02307*.

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.

Xiang Deng, Ahmed Hassan Awadallah, Christopher Meek, Oleksandr Polozov, Huan Sun, and Matthew Richardson. 2020. Structure-grounded pretraining for text-to-sql. *arXiv preprint arXiv:2010.12773*.

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*, pages 124–133.

Hamza Harkous, Isabel Groves, and Amir Saffari. 2020. Have your text and use it too! end-to-end neural data-to-text generation with semantic fidelity. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2410–2424, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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

Svetlana Kiritchenko and Saif M Mohammad. 2017. Capturing reliable fine-grained sentiment associations by crowdsourcing and best-worst scaling. *arXiv preprint arXiv:1712.01741*.

Oliver Lehmberg, Dominique Ritze, Robert Meusel, and Christian Bizer. 2016. A large public corpus of web tables containing time and context metadata. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 75–76.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Liang Li, Can Ma, Yinliang Yue, Linjun Shou, and Dayong Hu. 2020. Learning better representation for tables by self-supervised tasks. *arXiv preprint arXiv:2010.07606*.

Liunian Li and Xiaojun Wan. 2018. Point precisely: Towards ensuring the precision of data in generated texts using delayed copy mechanism. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1044–1055.

Percy Liang, Michael I 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: Volume 1-Volume 1*, pages 91–99. 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 *Thirty-Second AAAI Conference on Artificial Intelligence*.

Jordan J Louviere, Terry N Flynn, and Anthony Alfred John Marley. 2015. *Best-worst scaling: Theory, methods and applications*. Cambridge University Press.

Manuel Mager, Ramón Fernandez Astudillo, Tahira Naseem, Md Arafat Sultan, Young-Suk Lee, Radu Florian, and Salim Roukos. 2020. GPT-too: A language-model-first approach for AMR-to-text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1846–1852, Online. Association for Computational Linguistics.

Hongyuan Mei, Mohit Bansal, and Matthew R Walter. 2015. What to talk about and how? selective generation using lstms with coarse-to-fine alignment. *arXiv preprint arXiv:1509.00838*.

Xiaochang Peng, Chuan Wang, Daniel Gildea, and Nianwen Xue. 2017. Addressing the data sparsity issue in neural amr parsing. *arXiv preprint arXiv:1702.05053*.

Laura Perez-Beltrachini and Mirella Lapata. 2018. Bootstrapping generators from noisy data. *arXiv preprint arXiv:1804.06385*.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with entity modeling. *arXiv preprint arXiv:1906.03221*.

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 preprint arXiv:1910.10683*.

Clément Rebuffel, Laure Soulier, Geoffrey Scoutheeten, and Patrick Gallinari. 2020. A hierarchical model for data-to-text generation. In *European Conference on Information Retrieval*, pages 65–80. Springer.

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

Leonardo FR Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2020. Investigating pretrained language models for graph-to-text generation. *arXiv preprint arXiv:2007.08426*.

Marco Roberti, Giovanni Bonetta, Rossella Cancelliere, and Patrick Gallinari. 2019. Copy mechanism and tailored training for character-based data-to-text generation. *arXiv preprint arXiv:1904.11838*.

Lei Sha, Lili Mou, Tianyu Liu, Pascal Poupart, Sujian Li, Baobao Chang, and Zhifang Sui. 2018. Order-planning neural text generation from structured data. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Anastasia Shimorina and Claire Gardent. 2018. Handling rare items in data-to-text generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 360–370, Tilburg University, The Netherlands. Association for Computational Linguistics.

Amanda Stent, Rashmi Prasad, and Marilyn Walker. 2004. Trainable sentence planning for complex information presentation in spoken dialog systems. In *Proceedings of the 42nd annual meeting on association for computational linguistics*, page 79. Association for Computational Linguistics.

Marilyn A Walker, Amanda Stent, François Mairesse, and Rashmi Prasad. 2007. Individual and domain adaptation in sentence planning for dialogue. *Journal of Artificial Intelligence Research*, 30:413–456.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*.

Sam Wiseman, Stuart M Shieber, and Alexander M Rush. 2017. Challenges in data-to-document generation. *arXiv preprint arXiv:1707.08052*.

Sam Wiseman, Stuart M Shieber, and Alexander M Rush. 2018. Learning neural templates for text generation. *arXiv preprint arXiv:1808.10122*.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. Tabert: Pretraining for joint understanding of textual and tabular data. *arXiv preprint arXiv:2005.08314*.