gTBLS: Generating Tables from Text by Conditional Question Answering

Anonymous ACL submission

Abstract

Distilling large, unstructured text into a structured, condensed form such as tables is an open research problem in Natural Language Processing. Prior approaches address this task through additional parameters in the Transformer's attention mechanism. This paper presents Generative Tables (gTBLS), a two-stage, parameter-efficient solution to automatically construct structured tables from text. The first stage infers table structure (row and column headers) from the text, and the second stage formulates questions and fine-tunes a causal language model to answer them. gTBLS improves prior approaches by up to 21% in BERTScore on the table content generation task of the E2E, WikiTableText, WikiBio, and Rotowire datasets with 66% fewer parameters.

1 Introduction

An important challenge in Natural Language Processing is summarization, distilling large, unstructured texts into a condensed form while preserving factual consistency. There has been substantial work summarizing news articles, medical information, and conversational dialogue (Nallapati et al., 2016; See et al., 2017; Shang et al., 2018; Joshi et al., 2020; Chen and Yang, 2020). However, these efforts focus on transforming unstructured text into shorter yet unstructured forms. Compiling unstructured knowledge sources into structured forms such as tables remains an open research problem.

Organizing information into tables provides several advantages compared to unstructured paragraphs (Tang et al., 2023). Tabular information is more efficient, utilizing row and column headers to reduce redundancy. Additionally, the structured presentation simplifies the task of comparing different sources of information, especially when dealing with quantitative data. However, manually creating tables from text is time-consuming and error-prone.

Driven by the success of Large Language Models (LLMs) on sequence-to-sequence natural language tasks, recent work explored the automatic generation of structured knowledge from unstructured text (Wu et al., 2022; Pietruszka et al., 2022). A primary challenge in this automatic table generation task lies in ensuring their syntactic validity. Every row and column in a table must contain the same number of cells, with row and column headers delineating relationships between cells. Prior work addresses this constraint by including additional parameters like row and column relation embeddings (Wu et al., 2022) or positional bias (Pietruszka et al., 2022).

In contrast, we propose Generative Tables (gTBLS) [1], a novel, two-stage, parameter-efficient approach to condense unstructured textual informa-

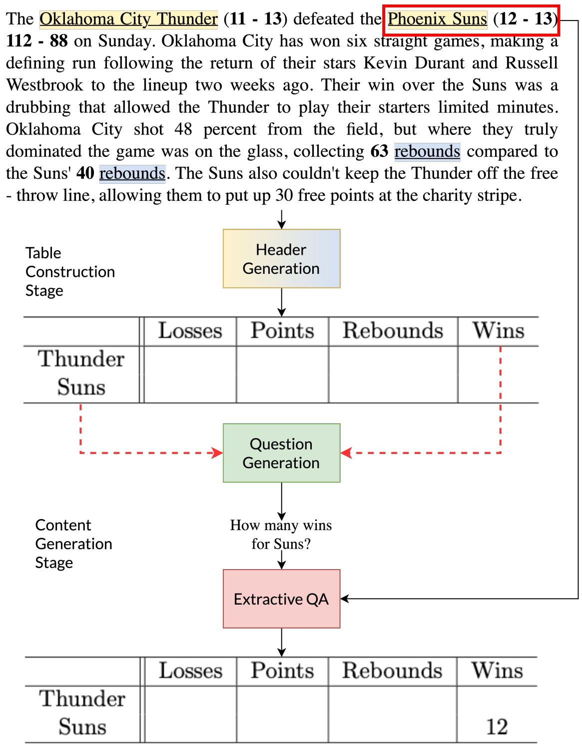

Figure 1: Overview of Generative Tables (gTBLS). gTBLS uses a two stage, parameter-efficient approach to condense textual information into structured tables.

[1] Our code will be released with the camera-ready version

tion into structured tables. The first stage infers table structure (row and column headers) from text. The second stage uses the generated headers to formulate questions. A causal language model is then fine-tuned to answer these questions using the textual paragraph as evidence. An overview of gTBLS is provided in Figure 1. gTBLS achieves similar or better performance than previous approaches while using up to 66% fewer parameters on the E2E, WikiTableText, WikiBio, and Rotowire datasets.

Section 2 presents an overview of related work that addresses the challenge of generating structured content from textual paragraphs. Section 3 describes our novel approach, which frames table generation as conditional question answering. Section 4 describes the dataset, experimental procedure, and results. We conclude the paper in Section 5 and outline limitations in Section 6.

2 Related Work

Early research on tabular generation focused on discriminative techniques. Branavan et al. (2007) used a tree-based method to infer a table of contents from documents, while Aramaki et al. (2009) treated tabular generation as a multi-label classification problem, with predefined headers.

More recent neural approaches for table generation have utilized Generative Adversarial Networks (GANs) to synthesize tabular data from existing datasets (Xu and Veeramachaneni, 2018; Park et al., 2018; Chen et al., 2019). Similarly, research in generating structured information, like knowledge graphs and entities from text, has also been explored (Hakkani-Tür et al., 2013; Luan et al., 2018; Deng et al., 2021; Lu et al., 2022).

Recent work directly addressing text-to-table generation includes (Wu et al., 2022), (Pietruszka et al., 2022), and (Tang et al., 2023). Wu et al. (2022) proposed modifying the Transformer decoder's attention mechanism, incorporating row and column relation embeddings to capture header and non-header cell relationships. Pietruszka et al. (2022) utilize learnable bias parameters to encode relative cell positions. Finally, Tang et al. (2023) employed structure-aware instruction-tuning to fine-tune LLMs to generate tables.

In contrast, our approach, gTBLS, uses a two-stage process splitting the task into table structure construction and table content generation to capture inter-cell relationships and adhere to tabular constraints while maintaining parameter efficiency.

3 Table Generation as Question Answering

The foundation of Generative Tables (gTBLS) is a two-stage approach to table generation that disentangles structure generation and information retrieval. While LLMs have demonstrated success on text generation and information retrieval independently, utilizing them to generate structured knowledge is more complex. Rows and columns impose structure requirements during inference. LLM-based methods that generate tables sequentially (e.g., row-by-row or column-by-column) face a critical challenge: the number of cells generated in the initial row or column determines the structure of the entire table. Failing to adhere to these constraints results in structurally invalid tables. gTBLS addresses this issue by first employing a Table Construction stage to identify row and column headers from natural language text to construct an empty table with headers. Then, the Table Content Generation stage uses the identified headers to fill cell contents with synthetically generated QA pairs, ensuring the validity of all generated tables. The process is described in Figure 1.

Table Construction: In this stage, gTBLS utilizes a causal language model to generate row and column headers. During training, the model extracts row and column headers from ground-truth tables. These headers are concatenated sequentially and separated by a <SEP> token. The Table Construction stage is framed as conditional text generation, where the causal language model is fine-tuned to generate the concatenated header sequence conditioned on the textual paragraph. During inference, the model generates a sequence of headers based solely on the textual input.

Table Content Generation: This stage synthetically generates QA fine-tuning pairs over the skeleton of the Table constructed in the previous stage. Using the generated rows and columns from above, gTBLS formulates a question, the answer to which is the cell content. A separate question is formulated for each combination of row and column header in the format 'What is the {Column value} for {Row value}?'. For example, given the row header 'Suns' and the column header 'Wins', the formulated question is 'What is the number of Wins for Suns?'. A LLM is fine-tuned to answer this question using the textual input as evidence.

Dataset	Model	Header Cell F1	Header Cell BERTScore	Non-Header Cell F1	Non-Header Cell BERTScore	Params
E2E	Wu et al.	**99.63**	**99.88**	97.94	98.57	406M
	gTBLS	**99.63**	99.85	**98.11**	**99.46**	**247M**
WikiTableText	Wu et al.	**78.16**	**95.68**	62.71	80.74	406M
	gTBLS	71.34	92.90	**65.34**	**93.50**	**247M**
Wikibio	Wu et al.	**80.52**	**92.60**	**69.71**	76.56	406M
	gTBLS	75.33	92.50	65.75	**92.53**	**247M**

Table 1: Comparison between the performance of Generative Tables (gTBLS) and the prior state of the art introduced by Wu et al. (2022). gTBLS uses almost 40% fewer parameters than Wu et al. (2022)

Dataset	Model	Row Header F1	Row Header BERTScore	Column Header F1	Column Header BERTScore	Non-Header Cell F1	Non-Header Cell BERTScore	Params
Rotowire Team	Wu et al.	94.97	97.51	86.02	89.05	86.31	90.80	406M
	STable	94.97	**97.80**	**88.90**	88.70	84.70	90.30	737M
	gTBLS	94.95	**97.79**	82.89	**94.99**	**86.38**	**96.37**	**247M**
Rotowire Player	Wu et al.	92.31	93.71	87.78	94.41	**86.83**	88.97	406M
	STable	**93.50**	**95.10**	**88.10**	94.50	84.50	90.40	737M
	gTBLS	88.68	**95.07**	83.76	**96.02**	84.42	**95.03**	**247M**

Table 2: Comparison between prior approaches Wu et al. (2022), STable (Pietruszka et al., 2022), and gTBLS. Rotowire is a more challenging dataset since tables contain multiple rows and columns when compared to the datasets in Table 1 which contain multiple rows but only a single column.

4 Experimental Results

4.1 Text-to-Table Datasets

Wu et al. (2022) propose four datasets for the text-to-table task by inverting datasets created for the dual problem of generating textual descriptions from tables. Each dataset consists of textual paragraphs and paired tabular information summarizing content in the text.

The E2E dataset (Novikova et al., 2017) concerns restaurant descriptions, requiring summarization of information into tables with descriptors like restaurant name, customer rating, and location. WikiTableText (Bao et al., 2018), sourced from Wikipedia, consists of natural language descriptions generated from tabular data across various topics. WikiBio (Lebret et al., 2016) comprises introductions of individuals from Wikipedia alongside tabular summaries extracted from the same page's information box. Rotowire (Wiseman et al., 2017) presents NBA game reports and two separate tables summarizing team and player statistics. While E2E, WikiTableText, and WikiBio contain tables with multiple rows and a single column, Rotowire is a more challenging dataset with multi-row, multi-column tables, necessitating strict adherence to equal cell counts across rows and columns.

4.2 Procedure

Training: We fine-tune FlanT5-base (Chung et al., 2022) separately for Table Construction and Table Content Generation. All experiments fine-tune for 10 epochs with AdamW (Loshchilov and Hutter) and a learning rate of 5e-5 on 8 Nvidia A40 GPUs.

Evaluation: We report F1 and BERTScore. The F1 score is the harmonic mean of precision and recall of predicted cells compared to the ground truth. We ignore cells with empty content in this computation. BERTScore (Zhang* et al., 2020) measures token similarity between candidate and reference sentences through contextual embeddings, and captures semantic similarity. A detailed description of these metrics is available in (Wu et al., 2022).

4.3 Results

Tables 1 and 2 present a comparative analysis of the performance of gTBLS with Text-to-Table (Wu et al., 2022) and STable (Pietruszka et al., 2022) representing the current state-of-the-art (SoTA). Table 1 displays results on the E2E, WikiTableText,

Dataset	Header F1		Cell F1		Error Rate	
	Seq2Seq	gTBLS	Seq2Seq	gTBLS	Seq2Seq	gTBLS
E2E	99.60	**99.63**	97.94	**98.11**	**0.0%**	**0.0%**
WikiTableText	69.71	**71.34**	**66.61**	65.34	0.6%	**0.0%**
Wikibio	**76.36**	75.33	63.51	**65.75**	1.64%	**0.0%**
Rotowire Team	57.84	**88.92**	51.18	**86.38**	30.9%	**0.0%**
Rotowire Player	26.34	**86.22**	12.80	**84.42**	57.28%	**0.0%**

Table 3: Comparison of F1 scores between sequence to sequence baseline and gTBLS

Dataset	gTBLS-TC	Ground Truth-TC
E2E	98.11	98.50
WikiTableText	65.34	70.73
Wikibio	65.67	77.06
Rotowire Team	72.27	95.18
Rotowire Player	52.75	88.15

Table 4: Comparison between gTBLS' Content Generator using row and column headers generated by gTBLS' Table Constructor (TC) versus ground-truth (F1 scores)

and WikiBio datasets. Table 2 shows results with Rotowire Team and Rotowire Player datasets.

For Table Construction (Header Cell) in Table 1, despite employing nearly 40% fewer parameters and having half the context length, gTBLS achieves an F1 score that matches the SoTA for the E2E dataset and is within 9% and 6.5% (relative) on the WikiTableText and Wikibio datasets, respectively. On these same datasets, gTBLS is within 3% on WikiTableText with BERTScore and achieves parity for the other two.

For Table Content Generation (Non-Header Cell) in Table 1, gTBLS outperforms the SoTA in all three datasets in BERTScore (1%, 16%, and 21% improvements across E2E, WikiTableText, and Wikibio, respectively) and improves F1 on two of the three datasets. For context, Text-to-Table's BART-large (Lewis et al., 2020) backbone employs 406M parameters and a context length of 1024, whereas gTBLS uses 247M parameters and a context length of 512.

Table 2 presents results on the Rotowire Team and Player datasets. The Rotowire datasets present a more difficult challenge for table generation than E2E, WikiTableText, and WikiBio due to the more complex table structure containing multiple rows and columns. For Table Construction (Row/Column Header), gTBLS outperforms the SoTA on BERTScore by 7.1% and 1.6% (relative) on Rotowire Team and Player, respectively, and achieves competitive F1 scores (within 5%) of the SoTA STable (Pietruszka et al., 2022) while using almost 66% fewer parameters. For Table Content Generation (Non-Header Cell), gTBLS outperforms the SoTA by 5-6% on BERTScore and is within 2.7% on F1.

In Table 3, we compare gTBLS with a sequence-to-sequence approach that models table generation as conditional generation of a flattened table representation. The '|' token separates column values and a <NEWLINE> tag separates rows. We report the mean of row and column header F1s for the Rotowire datasets. gTBLS achieves up to a 2.3% and 3.5% relative improvement in F1 scores for Table Construction and Table Content Generation tasks, respectively. Notably, on the Rotowire datasets, gTBLS excels, consistently generating valid tables while the sequence-to-sequence approach exhibits an error rate exceeding 50% on the Rotowire Player dataset. gTBLS ensures the reliability of all generated tables through its two-stage process.

Finally, Table 4 evaluates gTBLS' Content Generator. The F1 score for Table Content Generation using ground-truth headers surpasses that with generated headers, representing the upper bound assuming ideal header generation.

5 Conclusion

This paper introduces Generative Tables (gTBLS), a parameter-efficient approach to generate tables from text. gTBLS uses a two-stage process, first constructing a tabular structure using a causal language modeling objective followed by question answering to fill in the content. A key advantage of the two-stage approach is that all tables generated by gTBLS are valid without requiring post-processing. gTBLS improves prior approaches by up to 21% in BERTScore and achieves overall parity in F1 on the table content generation task of the E2E, WikiTableText, WikiBio, and Rotowire datasets with 66% fewer parameters.

6 Limitations

The gTBLS method, though effective as a parameter-efficient approach for table generation from text, presents unresolved challenges. First, its performance is limited by the context length of the utilized models, leading to the omission of header and cell information from later parts of the source text. Additionally, its reliance on generating question-answer pairs from row and column headers restricts it to tables with a direct header-cell correlation. Complex table structures, like headers spanning multiple rows or columns, remain a challenge. Moreover, gTBLS is optimized for generating dense tables, where cell content directly corresponds to the text. This study excludes cells without matching text information to align with evaluation frameworks proposed by prior work. However, future approaches could explore generating sparse tables, potentially incorporating unknown <UNK> tokens as needed.

References

Eiji Aramaki, Yasuhide Miura, Masatsugu Tonoike, Tomoko Ohkuma, Hiroshi Mashuichi, and Kazuhiko Ohe. 2009. TEXT2TABLE: medical text summarization system based on named entity recognition and modality identification. In *Proceedings of the Workshop on BioNLP - BioNLP '09*, page 185, Boulder, Colorado. Association for Computational Linguistics.

Junwei Bao, Duyu Tang, Nan Duan, Zhao Yan, Yuanhua Lv, Ming Zhou, and Tiejun Zhao. 2018. Table-to-Text: Describing Table Region with Natural Language. ArXiv:1805.11234 [cs].

S. R. K. Branavan, Pawan Deshpande, and Regina Barzilay. 2007. Generating a table-of-contents. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 544–551, Prague, Czech Republic. Association for Computational Linguistics.

Haipeng Chen, Sushil Jajodia, Jing Liu, Noseong Park, Vadim Sokolov, and V. S. Subrahmanian. 2019. FakeTables: Using GANs to Generate Functional Dependency Preserving Tables with Bounded Real Data. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 2074–2080, Macao, China. International Joint Conferences on Artificial Intelligence Organization.

Jiaao Chen and Diyi Yang. 2020. Multi-view sequence-to-sequence models with conversational structure for abstractive dialogue summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4106–4118, Online. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling Instruction-Finetuned Language Models. ArXiv:2210.11416 [cs].

Xiang Deng, Ahmed Hassan Awadallah, Christopher Meek, Oleksandr Polozov, Huan Sun, and Matthew Richardson. 2021. Structure-grounded pretraining for text-to-SQL. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1337–1350, Online. Association for Computational Linguistics.

Dilek Hakkani-Tür, Asli Celikyilmaz, Larry Heck, and Gokhan Tur. 2013. A weakly-supervised approach for discovering new user intents from search query logs. In *Interspeech 2013*, pages 3780–3784. ISCA.

Anirudh Joshi, Namit Katariya, Xavier Amatriain, and Anitha Kannan. 2020. Dr. summarize: Global summarization of medical dialogue by exploiting local structures. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3755–3763, Online. Association for Computational 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*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.

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

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772, Dublin, Ireland. Association for Computational Linguistics.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. ArXiv:1808.09602 [cs].

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E Dataset: New Challenges For End-to-End Generation. ArXiv:1706.09254 [cs].

Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. 2018. Data synthesis based on generative adversarial networks. *Proceedings of the VLDB Endowment*, 11(10):1071–1083.

Michał Pietruszka, Michał Turski, Łukasz Borchmann, Tomasz Dwojak, Gabriela Pałka, Karolina Szyndler, Dawid Jurkiewicz, and Łukasz Garncarek. 2022. STable: Table Generation Framework for Encoder-Decoder Models. ArXiv:2206.04045 [cs].

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 (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Guokan Shang, Wensi Ding, Zekun Zhang, Antoine Tixier, Polykarpos Meladianos, Michalis Vazirgiannis, and Jean-Pierre Lorré. 2018. Unsupervised abstractive meeting summarization with multi-sentence compression and budgeted submodular maximization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 664–674, Melbourne, Australia. Association for Computational Linguistics.

Xiangru Tang, Yiming Zong, Jason Phang, Yilun Zhao, Wangchunshu Zhou, Arman Cohan, and Mark Gerstein. 2023. Struc-Bench: Are Large Language Models Really Good at Generating Complex Structured Data? ArXiv:2309.08963 [cs].

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Xueqing Wu, Jiacheng Zhang, and Hang Li. 2022. Text-to-table: A new way of information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2518–2533, Dublin, Ireland. Association for Computational Linguistics.

Lei Xu and Kalyan Veeramachaneni. 2018. Synthesizing Tabular Data using Generative Adversarial Networks. ArXiv:1811.11264 [cs, stat].

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.

A Appendix

We acknowledge the use of GitHub Copilot to assist in code completion.