Unifying Structured Data as Graph for Data-to-Text Pre-Training

Shujie Li[1*] **Liang Li**[3] **Ruiying Geng**[2] **Min Yang**[1†] **Binhua Li**[2] **Guanghu Yuan**[1]
Wanwei He[1] **Shao Yuan**[2] **Can Ma**[3] **Fei Huang**[2] **Yongbin Li**[2†]
[1]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, China
[2]DAMO Academy, Alibaba Group, China
[3]Institute of Information Engineering, Chinese Academy of Sciences, China
{sj.li1, min.yang}@siat.ac.cn, shuide.lyb@alibaba-inc.com

Abstract

Data-to-text (D2T) generation aims to transform structured data into natural language text. Data-to-text pre-training has proved to be powerful in enhancing D2T generation and yields impressive performance. However, previous pre-training methods either oversimplified structured data into a sequence without considering input structures or designed training objectives tailored for a specific data structure (e.g., table or knowledge graph). In this paper, we unify different types of structured data (i.e., table, key-value data, knowledge graph) into the graph format and cast different D2T generation tasks as graph-to-text generation. To effectively exploit the structural information of the input graph, we propose a structure-enhanced pre-training method for D2T generation by designing a structure-enhanced Transformer. Concretely, we devise a position matrix for the Transformer, encoding relative positional information of connected nodes in the input graph. In addition, we propose a new attention matrix to incorporate graph structures into the original Transformer by taking the available explicit connectivity structure into account. Extensive experiments on six benchmark datasets show the effectiveness of our model. Our source codes are available at https://github.com/AlibabaResearch/DAMO-ConvAI/tree/main/unid2t.

1 Introduction

Data-to-text (D2T) generation, which aims to generate a target natural language text conditioned on source structured data, has attracted noticeable attention due to its many applications such as journalism (Rebuffel et al., 2020), medical diagnosis (Nishino et al., 2020), financial and weather reports (Liang et al., 2009), and sports broadcasting (Chen and Mooney, 2008). The input structured data can include tables of records, simulations of physical systems, spreadsheets, knowledge graphs, and so on. Transforming structured data into textual data can facilitate a wide range of users to understand and use the structured data, which is needed in many real-life scenarios.

Recently, large-scale pre-trained models have proved to be powerful in D2T generation and yield impressive performance (Kale and Rastogi, 2020; Xing and Wan, 2021; Liu et al., 2022), which benefit from the rich knowledge contained in large-scale pre-training corpora. Xing and Wan (2021) proposed a structure-aware table-to-text pre-training model, which devised three self-supervised training objectives tailored for modeling tables and their contexts. Ke et al. (2021) adopted a structure-aware semantic aggregation module to model the structure of an input graph at each Transformer layer, and explicitly learned graph-text alignments instead of directly fine-tuning text-to-text pre-trained models on graph-to-text corpora.

Although significant progress has been made in this field, there are still several technical challenges with existing D2T pre-training methods. Most prior studies made a cumbersome design tailored for a specific data structure such as tables (Liu et al., 2022) or knowledge graphs (Li et al., 2022), which could not effectively deal with diverse structured data in a unified framework. Kale and Rastogi (2020) were the first to study the ‘‘pre-train and fine-tune’’ strategy on several benchmarks spanning task-oriented dialogue, table-to-text, and graph-to-text. However, they oversimplified the input structured data into a flat string and adopted an original Transformer without capturing the structural information of source structured data.

[*] Work done during an internship at Alibaba DAMO Academy.

[†] Corresponding authors.

Transactions of the Association for Computational Linguistics, vol. 12, pp. 210–228, 2024. https://doi.org/10.1162/tacl_a_00641
Action Editor: Benjamin Van Durme. Submission batch: 2/2023; Revision batch: 11/2023; Published 3/2024.

In this paper, we **uni**fy the structured data into the graph format for **d**ata-**t**o-**t**ext pre-training (denoted as **UniD2T**). We convert diverse types of structured data into a unified graph format, keeping the structural information of the structured data. We treat the items in the structured data as a set of nodes and connect the nodes according to the connectivity of the structured data. In this way, we can cast various D2T tasks as the graph-to-text generation task.

To effectively encode the graph structure, we propose a structure-enhanced pre-training model, which can be applied to various downstream D2T generation tasks. Our proposed D2T pre-training model is built upon the T5 model (Raffel et al., 2020). Since the T5 model is a text-to-text transfer Transformer framework and cannot effectively encode the graph structure, we propose a structure-enhanced Transformer to encode the structural information. Concretely, we propose an explicit position matrix for the Transformer, encoding the relative positional information of connected nodes in the input graph. In addition, we build a new attention matrix to replace the attention mask in self-attention of the original Transformer, which encodes graph structures and takes the available explicit connectivity structure into account.

Our main contributions are three-fold. (1) We unify diverse types of structured data into a graph format and cast all D2T tasks as the graph-to-text generation task taking a graph as input and producing a text as output. (2) We propose a structure-aware pre-training method for D2T generation based on the T5 model, which incorporates relative positional information and graph structures into the original Transformer via two new position and attention matrices, respectively. (3) We conduct extensive experiments on six D2T benchmarks and achieve substantially better performance than strong baselines. We believe that the release of our unified D2T pre-training model will advance the research in this area.

2 Related Works

2.1 Data-to-Text Generation

Data-to-text (D2T) generation aims to produce output texts from structured data and has attracted noticeable attention from the natural language processing (NLP) community (Reiter and Dale, 1997). Recently, neural D2T models (Song et al., 2018; Zhu et al., 2019) have been the mainstream for this task and made impressive progress. The end-to-end neural models generate text directly from structured data by using an encoder-decoder architecture (Sutskever et al., 2014). These studies usually focus on improving the encoder structures based on attention mechanisms (Koncel-Kedziorski et al., 2019; Mehta et al., 2022) or graph neural networks (GNNs) (Philipp and Schütze, 2021; Ribeiro et al., 2021a,b). For example, Wang et al. (2020) proposed a graph-to-sequence model using a pairwise interaction function to obtain semantic relations between concepts. Puduppully et al. (2022) suggested a neural architecture that incorporated a planning module to manage high-level information in a logical and meaningful manner. Liu et al. (2018) proposed a structure-aware sequence-to-sequence architecture, which incorporated the filed information as additional input to the table encoder. Song et al. (2018) introduced graph recurrent networks (GRNs) to encode the AMR nodes directly. Subsequently, Shi et al. (2020) proposed GNNs as the structural encoder, which updated the representations of nodes based on their immediate neighbors. To integrate both local and non-local features and learn a better structural representation of a graph, Guo et al. (2019) introduced dense connection and allowed deep GCNs. Different from the local information aggregation scheme, Cai and Lam (2020) proposed a graph transformer that used explicit relation encoding and allowed direct communication between two distant nodes.

2.2 Data-to-Text Pre-training Models

Recently, we have witnessed the remarkable success of pre-training methods in a wide range of NLP tasks (Kenton and Toutanova, 2019; Radford et al., 2018; Lan et al., 2019; Bi et al., 2020). Most pre-training models are initially designed to text-to-text generation, lacking the ability to encode structural information. Recently, there exist some pre-training models designed for D2T tasks (Chen et al., 2020b; Agarwal et al., 2021; Ke et al., 2021; Bai et al., 2022). For example, KGPT (Chen et al., 2020b) proposed a distantly supervised learning method to exploit large-scale unlabeled web text for data-to-text pre-training. However, these pre-training models consider only one specific data structure and cannot be applied to diverse downstream D2T tasks. Although Tang et al. (2022) proposed a multi-task supervised pre-training model

Statistics	PREDATA	DOWNDATA
# Datasets	2	6
# Instances	4,951,267	2,240,927
Avg. input tokens	84.1	63.7
Avg. target tokens	90.8	100.9
Avg. Nodes	17.8	19.4
Avg. Edges	112.3	103.1

Table 1: Statistics of our pre-training data.

(MVP) for a series of D2T generation tasks, they utilized the original Transformer to encode the linearized input data without considering the graph structures. UniLM (Dong et al., 2019) was a pre-trained universal language model, which incorporated modified self-attention masks to facilitate bidirectional encoding or unidirectional decoding. While UniLM offers the flexibility of bidirectional encoding, its encoding attention mask is designed primarily for processing unstructured text, thereby restricting its ability to capture the structural characteristics of input graphs.

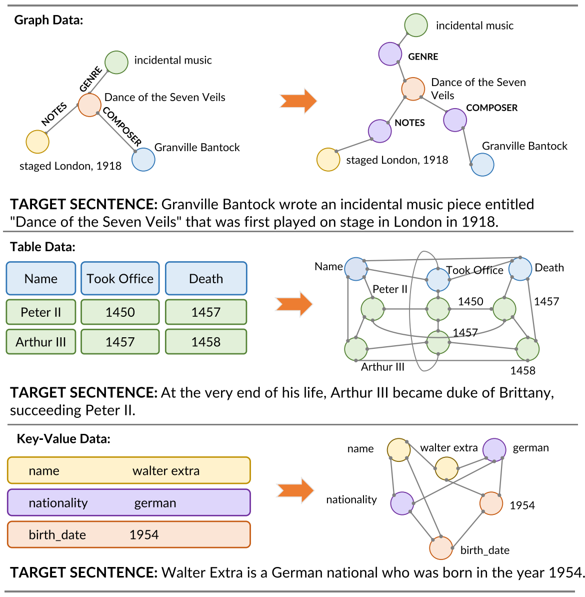
Different from previous work, we propose a unified pre-training model that casts all D2T tasks as the graph-to-text generation task. In addition, we incorporate graph structures into the original Transformer via two new position and attention matrices to effectively model the structured input data.

3 Pre-training Data Construction

Previous data-to-text pre-training datasets are usually tailored to specific structured data. In this paper, we collect eight D2T datasets from previous works and aggregate these datasets into a large corpus for pre-training our model. The statistics of pre-training data are provided in Table 1.

3.1 Existing Pre-training Datasets (PREDATA)

We first collect the table-text dataset TAPAS (Herzig et al., 2020) and the graph-text dataset KGTEXT (Chen et al., 2020b), which were originally designed for table-to-text and graph-to-text pre-training, respectively. TAPAS contains 6.2M tables from Wikipedia, and KGTEXT consists of 1.8M hyperlinked sentences from Wikipedia with the corresponding knowledge subgraphs from WikiData. We further devise a rule-based data-cleaning strategy to guarantee data quality. Finally, we obtain 4.9M data-text pairs (called PREDATA).



TARGET SECNTENCE: Granville Bantock wrote an incidental music piece entitled "Dance of the Seven Veils" that was first played on stage in London in 1918.

Name	Took Office	Death
Peter II	1450	1457
Arthur III	1457	1458

TARGET SECNTENCE: At the very end of his life, Arthur III became duke of Brittany, succeeding Peter II.

name	walter extra
nationality	german
birth_date	1954

TARGET SECNTENCE: Walter Extra is a German national who was born in the year 1954.

Figure 1: Unifying data in three formats into one graph structure.

3.2 Existing Downstream Datasets (DOWNDATA)

We also collect the training sets from six data-to-text datasets, including WebNLG (Gardent et al., 2017), DART (Nan et al., 2020), ToTTo (Parikh et al., 2020), WikiBio (Lebret et al., 2016), WikiTableT (Chen et al., 2021), and CoSQL (Yu et al., 2019). These datasets were designed for downstream data-to-text generation tasks. Concretely, WebNLG and DART are graph-to-text datasets; WikiBio and WikiTableT contain key-value pairs; ToTTo and CoSQL are table-based datasets. In total, there are about 2.2M instances (DOWNDATA). Notably, the test sets utilized for downstream tasks are expressly omitted from the pre-training data, ensuring the integrity of our experimental results by eliminating any potential data leakage.

3.3 Unifying Structured Data

As illustrated in Figure 1, we unify different structured data (knowledge graph, table, key-value pairs) into an unlabeled and connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that consists of a set of nodes $v \in \mathcal{V}$ and unlabeled edges $(v_i, v_j) \in \mathcal{V}$. Next, we elucidate the process of transforming the three distinct types of data (i.e., knowledge graphs, tables, and key-value pairs) into a unified graph $\mathcal{G}$.

(1) On the left side of Figure 1's Graph Data section, a knowledge graph can be formally expressed as $\mathcal{G}_0 = (\mathcal{V}_0, \mathcal{E}_0, \mathcal{R}_0)$, where nodes are denoted by $v \in \mathcal{V}_0$, and labeled edges are represented as $(v_s, r, v_t) \in \mathcal{E}_0$, with $r \in \mathcal{R}_0$ signifying the relation type. To more effectively model the relationships between nodes within the knowledge graph $\mathcal{G}_0$ without modifying the underlying model architecture, we transform it into its equivalent Levi graph, as shown on the right side of the Graph Data section in Figure 1, following similar methodologies as in prior studies (Ribeiro et al., 2021b; Li et al., 2022). A Levi graph is formally characterized as an unlabeled, connected bipartite graph, denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Specifically, each relation in $\mathcal{R}_0$ is treated as a new graph node within $\mathcal{G}$ and amalgamated with all nodes in $\mathcal{V}_0$ to form the comprehensive node set $\mathcal{V}$. Subsequently, each edge $(v_s, r, v_t) \in \mathcal{E}_0$ labeled with a relation type is converted into two unlabeled, undirected edges $(v_s, r), (r, v_t) \in \mathcal{E}$. In addition, for each unlabeled edge, corresponding reverse edges (r, v_s), (v_t, r) are introduced. For instance, given a labeled edge (*Dance of the Seven Veils*, *GENRE*, *incidental music*), this conversion results in four unlabeled edges (*Dance of the Seven Veils*, *GENRE*), (*GENRE*, *Dance of the Seven Veils*), (*GENRE*, *incidental music*), and (*incidental music*, *Dance of the Seven Veils*), constituting the final Levi graph $\mathcal{G}$.

(2) In the Table Data section of Figure 1, situated on the left side, Tabular data is conventionally structured with numerous cells organized based on their respective roles and interrelations. A table can be formally represented as $\mathcal{T} = v_{i,j} | i \in [1, N], j \in [1, M]$, where $v_{i,j}$ denotes a table cell, and N and M represent the number of rows and columns in the table, respectively. Inspired by recent studies (Wang et al., 2022; Li et al., 2023a), we use a heuristic rule to transform the tabular data into a unified graph $\mathcal{G}$ by introducing unlabeled edges between cells based on their roles and relationships. This structural transformation serves to maintain the invariance of the table content and proficiently articulate the relationships among cells in the table. More precisely, all cells within $\mathcal{T}$ are considered as graph nodes in $\mathcal{G}$, denoted as $\mathcal{V} = \mathcal{T}$. Furthermore, we establish the set of unlabeled edges $\mathcal{E}$ in accordance with two guiding principles. First, for any two cells $v_{i,j}$ and $v_{i,z}$ situated within the same row, we introduce a forward edge $(v_{i,j}, v_{i,z})$ along with a corresponding reverse edge $(v_{i,z}, v_{i,j})$ into $\mathcal{E}$. Second, for any two cells $v_{i,j}$ and $v_{i,z}$ located in the same column, we append a forward edge $(v_{i,j}, v_{z,j})$ and its corresponding reverse edge $(v_{z,j}, v_{i,j})$ to $\mathcal{E}$. For instance, contemplating the right Table Data section in Figure 1, the cell ''Arthur III'' is linked not only to cells ''1457'' and ''1458'' in the same row but also to cells ''Name'' and ''Peter II'' in the same column. This intentional configuration is based on empirical observations and insights gained from data analysis. However, we acknowledge that there exists room for further exploration and experimentation concerning diverse node connectivity settings in future research. Given that the ToTTo dataset exclusively generates text for highlighted data, only the highlighted cells are considered as nodes.

(3) For Key-Value data in Figure 1, both key and value are regarded as nodes within $\mathcal{V}$. In addition to the requisite connection edges linking each (key, value) pair (e.g., the connection between the key *name* and the value *walter extra*), we extend our connectivity framework to include connections among keys themselves (e.g., the connection between *nationality* and *birth_date*) and value themselves (e.g., the connection between *walter extra* and *gernman*), drawing inspiration from the graph construction methodology commonly employed in table data analysis. In line with tabular data, we introduce both forward and reverse edges for any connected nodes within $\mathcal{V}$.

To ensure clarity and context in the generated text, we introduce two specific prefixes before the actual input data: (1) A data-independent prefix that universally states ''describe the following data.'' (2) A data-specific prefix, tailored according to the nature and structure of the data at hand. We provide the data-specific prefixes for the three data structures in Table 2. For example, the triple ''$Jens\_Hartel$ | $club$ |$Berliner\_AK\_07$'' from the DART dataset will add the common prefix and its special prefix to form an input ''*[Prefix] describe the following data: [Prefix] The category of the DBpedia entities is: SportsTeam. [Node] Jens_Hartel [Node] club [Node] Berliner _AK_07*''. We simplify the data-independent and data-specific prefixes to ''[*Prefix-I*]'' and ''[*Prefix-S*]'', respectively. The final input sequence with connectivity information is shown in Figure 2.

Type	Dataset	Prefix-S
Table	ToTTo	The table page title is: A, The table section title is: B
	CoSQL	select A from B where C
Graph	DART	The source is: A
	WebNLG	The category of the entities is: A, The number of RDF triples is: B
Key-Value	WikiBio	The article title is: A
	WikiTableT	the document title is: A, the section title is: B

Table 2: The data-specific prefixes that are tailored for different types of data. Here, A, B, and C can be replaced by the content of specific samples.

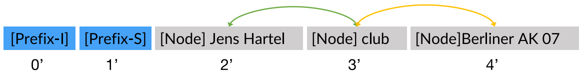


Figure 2: Simplified version of model input and connections between nodes.

4 Methodology

4.1 Problem Definition

We convert different structured data into a graph format and cast all data-to-text tasks as the graph-to-text (G2T) generation task. Formally, the G2T model takes a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ as input and produces a text $Y = \{y_1, \ldots, y_n\}$ as output, where $\mathcal{V}$ represents the entity set, $\mathcal{E}$ represents the relations between entities, and n is the length of the output text. Following previous studies (Ribeiro et al., 2020), we convert the graph $\mathcal{G}$ into an input sequence $\mathcal{G}_{\text{linear}} = \{x_1, \ldots, x_m\}$ consisting of m tokens.

4.2 Model Architecture

Our model is built upon the pre-trained T5 model given the impressive performance of T5 on text generation tasks. It is noteworthy that our pre-training strategy is model-agnostic and potentially applicable to any Transformer-based backbone networks. The encoder of Transformer is composed of a stack of blocks, each of which contains a self-attention layer followed by a feed-forward network. The decoder has a similar structure to the encoder except that it adopts a standard attention mechanism following a self-attention layer.

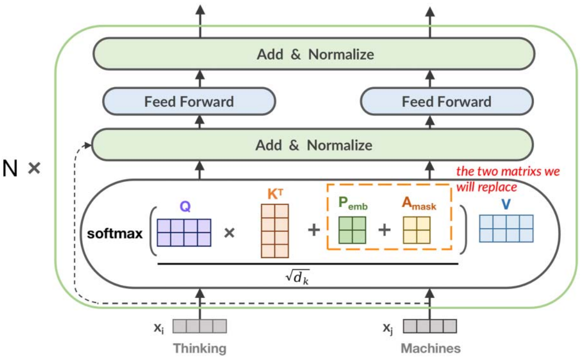


Figure 3: Transformer blocks on the T5-encoder side. The relative position and attention matrices in the self-attention calculation will be replaced by two novel position and attention matrices.

Preliminary In the case of the T5-encoder, a ‘‘fully-visible’’ attention mask is used, which permits the self-attention mechanism to consider all input entries when generating each output entry. In addition, T5 adopts a simplified form of position embeddings, where each embedding is a scalar. Formally, as illustrated in Figure 3, the attention calculation of encoder can be expressed as:

$$\mathbf{Q} = \mathbf{X}\mathbf{W}^Q, \mathbf{K} = \mathbf{X}\mathbf{W}^K, \mathbf{V} = \mathbf{X}\mathbf{W}^V \tag{1}$$

$$\alpha = \frac{1}{\sqrt{d}}\left(\mathbf{Q}\mathbf{K}^T + \mathbf{P}_{\text{emb}} + \mathbf{A}_{\text{mask}}\right) \tag{2}$$

$$\mathbf{Z} = \frac{\exp(\alpha)}{\sum \exp(\alpha)} \times \mathbf{V} \tag{3}$$

where $\mathbf{X}$ is the input sequence. $\mathbf{W}^Q \in \mathbb{R}^{d \times d_Q}$, $\mathbf{W}^K \in \mathbb{R}^{d \times d_K}$ and $\mathbf{W}^V \in \mathbb{R}^{d \times d_V}$ are learnable project parameters. α is the attention weight between the query vector $\mathbf{Q}$ and the key vector $\mathbf{K}$. d is the dimensionality of the hidden representations. $\mathbf{Z}$ is the output of the attention module. $\mathbf{P}_{\text{emb}}$ is position embedding and $\mathbf{A}_{\text{mask}}$ is attention mask.

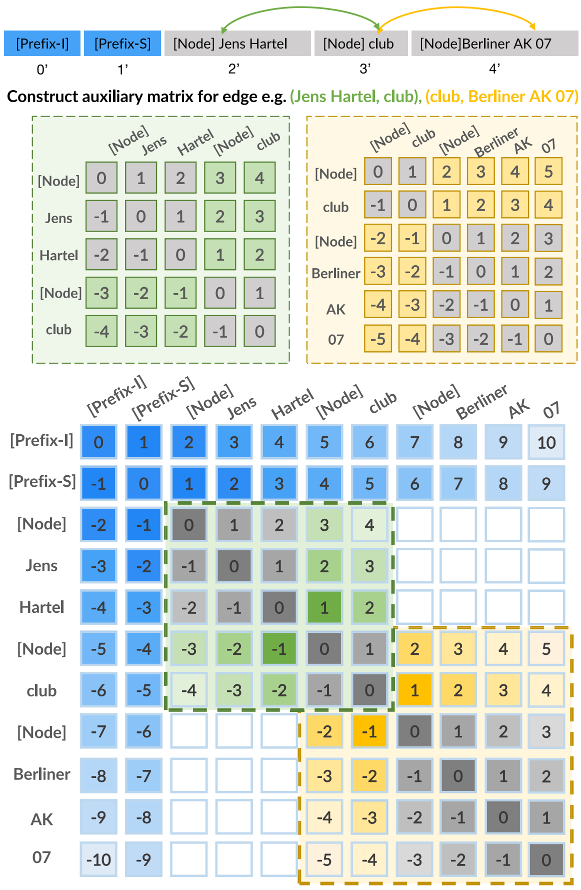
The original attention mechanism is designed to process unstructured natural language texts proves inadequate in effectively capturing the inherent structures within graphs. To better process our structured graph data, we replace the position embeddings $\mathbf{P}_{\text{emb}}$ and attention mask $\mathbf{A}_{\text{mask}}$ in Equation (2) with two new position and attention matrices respectively, ensuring their awareness of the underlying graph structures. Next, we will elaborate on the processes of constructing the position and attention matrices.

4.3 Structure-enhanced Transformer

T5 is based on an encoder-decoder Transformer, which does not necessarily capture graph structures. To address this issue, we propose a structure-enhanced Transformer, which is built upon the new position and attention matrices on the T5 encoder side. As illustrated in Figure 3, we use new position embedding and attention mask matrices (denoted as $\mathbf{P}_{\text{emb}}^{\text{new}}$ and $\mathbf{A}_{\text{mask}}^{\text{new}}$) to replace the $\mathbf{P}_{\text{emb}}$ and $\mathbf{A}_{\text{mask}}$ in the Equation (2), respectively. Specifically, we devise a position matrix for the Transformer to encode the relative positional information of connected nodes in the original input graph $\mathcal{G}$. In addition, we propose a new attention matrix to replace the attention mask in the self-attention, which takes the available explicit connectivity structure of the input graph into account.

4.3.1 Position Matrix Construction

Integrating relational information about the graph structure into the Transformer architecture is essential for graph-to-text generation. Nevertheless, most previous Transformer-based methods (Xing and Wan, 2021; Han and Shareghi, 2022) learned position embeddings automatically, instead of explicitly encoding the structural relationships. For the input graph, we should only consider the relative position between connected nodes but ignore the relative position between irrelevant nodes. To this end, we replace the positional embeddings of the original Transformer with a position matrix that only establishes the relative position between each relevant node pair (connected items). In this way, we can explicitly capture the relative positions of all relevant nodes precisely.



[Prefix-I]
[Prefix-S]
[Node] Jens Hartel
[Node] club
[Node]Berliner AK 07
0'
1'
2'
3'
4'
Construct auxiliary matrix for edge e.g. (Jens Hartel, club), (club, Berliner AK 07)

	[Node]	Jens	Hartel	[Node]	club
[Node]	0	1	2	3	4
Jens	-1	0	1	2	3
Hartel	-2	-1	0	1	2
[Node]	-3	-2	-1	0	1
club	-4	-3	-2	-1	0

	[Node]	club	[Node]	Berliner	AK	07
[Node]	0	1	2	3	4	5
club	-1	0	1	2	3	4
[Node]	-2	-1	0	1	2	3
Berliner	-3	-2	-1	0	1	2
AK	-4	-3	-2	-1	0	1
07	-5	-4	-3	-2	-1	0

	[Prefix-I]	[Prefix-S]	[Node]	Jens	Hartel	[Node]	club	[Node]	Berliner	AK	07
[Prefix-I]	0	1	2	3	4	5	6	7	8	9	10
[Prefix-S]	-1	0	1	2	3	4	5	6	7	8	9
[Node]	-2	-1	0	1	2	3	4				
Jens	-3	-2	-1	0	1	2	3				
Hartel	-4	-3	-2	-1	0	1	2				
[Node]	-5	-4	-3	-2	-1	0	1	2	3	4	5
club	-6	-5	-4	-3	-2	-1	0	1	2	3	4
[Node]	-7	-6				-2	-1	0	1	2	3
Berliner	-8	-7				-3	-2	-1	0	1	2
AK	-9	-8				-4	-3	-2	-1	0	1
07	-10	-9				-5	-4	-3	-2	-1	0

Figure 4: We construct a new position matrix $\mathbf{P}_{\text{emb}}^{\text{new}}$ to replace the original position matrix $\mathbf{P}_{\text{emb}}$ used in Equation (2). We first set an auxiliary matrix for each edge between two nodes, and then copy the content of the auxiliary matrix into the final position matrix. ***The distances of nodes lacking direct connections will be set to "±inf".*** The lighter the color, the farther the distance is.

Specifically, we first establish an auxiliary position matrix for each pair of connected nodes, similar to the green and yellow boxes in Figure 4. No matter how physically distant the two relevant nodes may be, the corresponding auxiliary position matrix solely takes into account the relative distance between these two nodes' internal tokens, disregarding the nodes situated between the two target nodes. For example, consider the input nodes "*[Node] club*" and "*[Node] Jens Hartel*", since "*club*" is 3 units to the right of "*Jens*", the value of cell [*Jens*, *club*] is 3. Notably, we only compute the relative distance between each connected note pair, while the distances of nodes lacking direct connections will be set to "$\pm inf$", signifying an infinite distance between them. For instance, the value assigned to the cell [*Jens*, *Berliner*] is "$+inf$" due to the absence of a direct connection between "*[Node] Jens Hartel*" and "*[Node] Berliner AK 07*".

After obtaining the auxiliary position matrix for each pair of connected items, we can construct the position matrix for the entire input sequence by copying the cell values from the corresponding auxiliary position matrices. It is noteworthy that we seek to endow the prefixes (denoted as "[*Prefix-I*]" and "[*Prefix-S*]") embedded within the input with the capacity to encapsulate comprehensive global information. Therefore, we postulate that these prefixes establish direct connections with other nodes within the input. Finally, we replace the positional embeddings $\mathbf{P}_{\text{emb}}$ of original Transformer with the

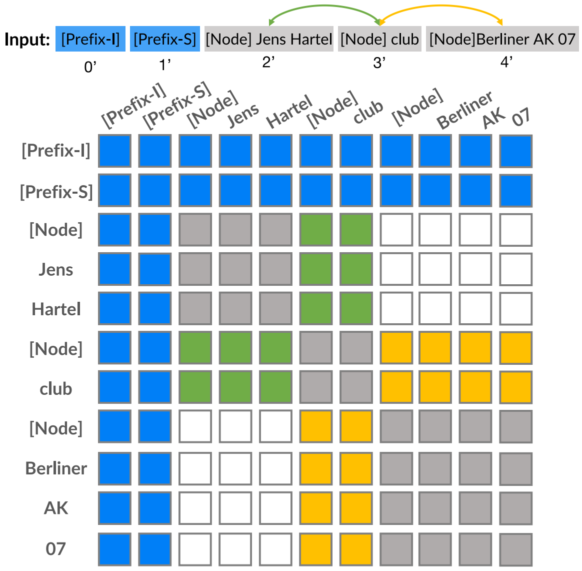


Figure 5: We construct a new attention matrix $\mathbf{A}_{\text{mask}}^{\text{new}}$ to replace the attention mask $\mathbf{A}_{\text{mask}}$ used in Equation (2). The attention matrix used to replace the attention mask of self-attention in Transformer. ***The values of the cells with colors are set to 1, while the values of the cells without colors are set to 0***. The blue color represents global attention, the gray color represents the self-connection of nodes, and the green and yellow colors represent the two connected edges.

learned position matrix $\mathbf{P}_{\text{emb}}^{\text{new}}$, so as to effectively capture the explicit relative distance between each pair of connected items.

4.3.2 Attention Matrix Construction

The self-attention in the original Transformer processes the input sequence by transforming the input sequence through the substitution of each element with a weighted average. Without refining the conventional attention mechanism, the present input data would be perceived as a fully interconnected graph, potentially hindering the optimal extraction of inherent structural information. Given the above reasons, we construct a relation-aware attention matrix to replace the original attention mask in self-attention. Concretely, if two elements have a direct relationship, we set the value of the corresponding cell to 1; otherwise, the value is set to 0. For example, as illustrated in Figure 5, since the items ‘‘*Jens Hartel*’’ and ‘‘*club*’’ have direct connection, the values of cells (*Jens*, *club*) and (*Hartel*, *club*) are set to 1; while since ‘‘*Jens Hartel*’’ and ‘‘*Berliner Ak 07*’’ have no direct connection, the values of the corresponding cells such as (*Jens*, *Berliner*) and (*Jens*, *AK*) are set to 0. Here, we hope that the prefixes (i.e., ‘‘[*Prefix-I*]’’ and ‘‘[*Prefix-S*]’’) within the input can carry global information, thus we make the prefixes attend to all other elements. After obtaining the attention matrix (denoted as $\mathbf{A}_{\text{mask}}^{\text{new}}$), we replace the attention matrix $\mathbf{A}_{\text{mask}}$ of self-attention in Equation (2) with our new attention matrix $\mathbf{A}_{\text{mask}}^{\text{new}}$ so as to effectively capture the graph structures as shown in Figure 3.

4.4 Pre-training Objectives

Similar to Andrejczuk et al. (2022), we first use the publicly available T5 checkpoints provided by Herzig et al. (2020) as the initialization. Then, we pre-train our model on our pre-training data. We employ two objectives to pre-train our model in a multi-task learning paradigm, including struct denoising and text generation objectives. In Table 3, we provide two specific training instances (input and output pairs) for the struct denoising and graph-to-text generation objectives.

Struct Denoising Objective We design a struct denoising strategy for table-like data, following the method used in T5, by training the model to predict a target sequence containing the missing or corrupted tokens in the input graph. We apply a noise function to construct a noisy input graph. In particular, the noise function is implemented by masking 15% of nodes while maintaining related edges in the graph. The goal of struct denoising objective is to reconstruct the target output that contains all the dropped-out nodes, delimited by the sentinel token. This pre-training objective helps the UniD2T model capture relationships between neighboring nodes in the input graph.

Graph-to-Text Generation Objective Given the linearized graph $\mathcal{G}_{\text{linear}}$ and its explicit connectivity structure $\mathcal{E}$, the graph-to-text generation task is carried out to produce the appropriate text to describe the given graph in an auto-regressive manner. We adopt the standard negative log-likelihood loss $\mathcal{L}_{\text{TG}}$ for the graph-to-text generation task:

$$\mathcal{L}_{\text{TG}} = -\frac{1}{N}\sum_{i=1}^{n} \log p(y_i|y_1,\ldots,y_{i-1};\mathcal{G}_{\text{linear}},\mathcal{E}) \quad (4)$$

where n is the length of the target sequence Y.

Task	Inputs	Targets
Struct Denoising	The category of the DBpedia entities is: $< extra\_id_0 >$. ‘Bakewell pudding’, ‘dish variation’, ‘$< extra\_id_1 >$’, ‘main ingredients’, ‘Ground almond, jam, butter, eggs’	$< extra\_id_0 >$ Food $< extra\_id_1 >$ Bakewell tart
Graph-to-Text Generation	Describe the following data: The category of the DBpedia entities is: Food. ‘Bakewell pudding’, ‘dish variation’, ‘Bakewell tart’, ‘main ingredients’, ‘Ground almond, jam, butter, eggs’	Bakewell tart is a variation of Bakewell pudding and some of the main ingredients are ground almonds, jam, butter and eggs.

Table 3: The examples of input-output pairs for struct denoising and graph-to-text generation objectives.

Dataset	Train	Valid	Test
ToTTo	120,761	7,700	7,700
CoSQL	7,845	1,074	–
WebNLG	13,211	1,667	1,779
DART	62,659	6,980	12,552
WikiBio	582,657	72,831	72,831
WikiTableT	1,453,794	4,533	4,351

Table 4: Statistics of downstream datasets.

5 Experimental Setup

5.1 Tasks and Datasets

To verify the generality and effectiveness of UniD2T, we conduct experiments on three types of data-to-text datasets. In particular, WebNLG (Gardent et al., 2017) and DART (Nan et al., 2020) are used for evaluating graph-to-text generation; WikiBio (Lebret et al., 2016) and WikiTableT (Chen et al., 2021) are utilized for evaluating key-value-to-text generation; ToTTo (Parikh et al., 2020) and CoSQL (Yu et al., 2019) are used for evaluating table-to-text generation. Table 4 provides the statistics of these six datasets.

5.2 Implementation Details

In the pre-training stage, our model is initialized with T5-Large. We pre-train our UniD2T model on NVIDIA A100 GPUs. The maximum sequence lengths of the input and target sequences are set to 1024 and 512, respectively. We set the batch size to 8. Gradient clipping is applied to the model with a maximum gradient value of 1. To alleviate the overfitting issue, the maximum number of training steps is 500k. Moreover, a patient step number is set to 25k, i.e., if the evaluation metrics does not increase for the patient step number, the training process will carry out an early stop. We set the maximum learning rate to 1e-5.

6 Experimental Results

6.1 Table-to-Text Generation

We conduct experiments on two table-to-text datasets, including ToTTo and CoSQL. The SQL queries within CoSQL and the table header information from ToTTo are strategically positioned within the data-specific prefixes, denoted as ‘‘[*Prefix-S*]’’, as illustrated in Table 2.

ToTTo ToTTo is an open-domain table-to-text task dataset that uses crowd annotators to highlight the table cells and revise the corresponding natural language descriptions. We compare our UniD2T with several strong baselines, including BERT2BERT (Rothe et al., 2020), LATTICE (Wang et al., 2022), CoNT (An et al., 2022), PlanGen (Su et al., 2021), and TABT5 (Andrejczuk et al., 2022). TABT5 is a pre-trained model tailored for table-to-text generation. We adopt BLEU (Papineni et al., 2002) and PARENT (Dhingra et al., 2019) as the evaluation metrics. The experimental results on ToTTo are summarized in Table 5. Our model achieves substantially better performance than the compared methods on ToTTo in terms of overall, overlap, and non-overlap settings. First, our model shows an improvement over T5 and TABT5, especially in terms of PARENT. Second, our model also achieves better results than the strong downstream methods.

CoSQL CoSQL serves as a prevalent benchmark for evaluating table-to-text models (Fang et al., 2022b; Li et al., 2023b). Each instance within CoSQL comprises an SQL query, the resultant table, and the corresponding response, where the SQL query gives explicit signals for models on what to generate. The generated description could provide a concise and easy-to-understand summary of the result table and help users verify whether the queried result is consistent with the

Models	Overall		Overlap		Non-Overlap	
	BLEU	PARENT	BLEU	PARENT	BLEU	PARENT
ChatGPT(gpt-3.5-turbo)	20.5	49.5	24.4	51.2	17.5	47.7
BERT-to-BERT(Rothe et al., 2020)	44.0	52.6	52.7	58.4	35.1	46.8
LATTICE (Wang et al., 2022)	48.4	58.1	56.1	62.4	40.4	53.9
CoNT (An et al., 2022)	49.1	58.9	56.7	63.2	41.3	54.6
PlanGen (Su et al., 2021)	49.2	58.7	56.9	62.8	41.4	54.2
T5-3B	49.5	58.4	57.5	62.6	41.4	54.2
TABT5 (Andrejczuk et al., 2022)	49.2	57.2	–	–	41.0	52.7
UniD2T	**49.9**	**59.8**	**57.8**	**64.0**	**42.0**	**55.7**

Table 5: Results on the ToTTo test set.

Models	BLEU	ROUGE-L
GraphWriter	16.86	47.44
FALCON	25.65	57.89
BART-Base	24.60	57.39
T5-Large	25.25	57.54
UniD2T	**32.68**	**61.47**

Table 6: Results on CoSQL development set.

original question. We compare our model with GraphWriter (Koncel-Kedziorski et al., 2019), BART-Base, T5-Large, and FALCON (Fang et al., 2022a) which is a faithful contrastive generation framework based on T5. We adopt BLEU (Papineni et al., 2002) and ROUGE-L (Lin, 2004) as evaluation metrics. Since CoSQL does not release the test set, we follow FALCON and report the experimental results on the development set in Table 6. Our UniD2T model achieves significantly better performance than baselines. The BLEU and ROUGE scores increase by 7.03 and 3.58, respectively, over the best-performing baseline FALCON.

6.2 Graph-to-Text Generation

We conduct experiments on two graph-to-text datasets, including DART and WebNLG.

DART DART is a large dataset for open-domain text generation that treats the input as a set of RDF entity-relation triples. We compare our UniD2T model with several pre-training models including Transformer, BART, T5, and the state-of-the-art method CONTROL PREFIXES (Clive et al., 2021). BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), and TER (Snover et al., 2005) are adopted as evaluation metrics.

Models	BLEU	METEOR	TER
End-to-End Transformer†	27.24	0.25	0.65
LSTM with Attention†	29.66	0.27	0.63
CONTROL PREFIXES	51.95	0.41	0.43
ChatGPT(gpt-3.5-turbo)	40.51	0.37	0.53
BART-Base†	47.11	0.38	0.46
BART-Large†	48.56	0.39	0.45
T5-Small†	47.69	0.39	0.46
T5-Base†	49.21	0.40	0.44
T5-Large†	50.66	0.40	0.43
UniD2T	**54.96**	**0.42**	**0.42**

Table 7: Evaluation results on DART test set. Results with † are token from DART (Nan et al., 2020).

As shown in Table 7, our model surpasses the best-performing model CONTROL PREFIXES by a 3.0% BLEU.

WebNLG WebNLG (Zhou and Lampouras, 2020) consists of a set of triples collected from DBpedia and the corresponding manually annotated text. BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), chrF++ (Popović, 2015), TER (Snover et al., 2005), and BLEURT (Sellam et al., 2020) are adopted as evaluation metrics. We compare our method with both pre-trained language models and strong downstream baselines. The overall experimental results on WebNLG are shown in Table 8. Our model achieves the highest performance among all baseline models, including the graph pre-training model TRIPLE (Han and Shareghi, 2022).

6.3 Key-Value-to-Text Generation

We conduct experiments on two key-value-based datasets, including WikiBio and WikiTableT.

Model	BLEU	METEOR	chrF++	TER	BLEURT
CP	54.97	41.7	69.3	39.8	0.62
CP + DART	55.41	41.9	69.8	39.2	0.63
T5-Large	51.74	40.3	66.9	41.7	0.61
TRIPLE	57.64	42.24	–	38.9	–
UniD2T	**60.41**	**44.35**	**73.4**	**34.1**	**0.65**

Table 8: Evaluation results on WebNLG test set. CP stands for CONTROL PREFIXES (Clive et al., 2021).

Models	WikiBio		WikiTableT	
	BLEU	PARENT	BLEU	PARENT
Transformer	44.3	74.0	19.5	42.8
SANA	45.7	76.9	–	–
CoNT	47.1	–	–	–
KGPT	45.1	76.3	31.8	48.5
T5-Large	48.6	77.5	31.4	47.6
UniD2T	**50.4**	**79.8**	**33.7**	**50.7**

Table 9: Results on WikiBio and WikiTableT test sets.

WikiBio WikiBio is designed to generate descriptions from a Wikipedia infobox and aims to generate the first sentence of a biography. We compare UniD2T with previous state-of-the-art model (i.e, CoNT [An et al., 2022]), pre-trained models (T5-Large, KGPT) and Non-autoregressive model SANA (Wang et al., 2021) on WikiBio. BLEU and PARENT are adopted as evaluation metrics. The results are reported in Table 9. UniD2T outperforms the best baseline CoNT by 3.7% on BLEU.

WikiTableT WikiTableT is collected from Wikipedia sections with their corresponding tabular data, which contains millions of instances. We compare UniD2T with Transformer, T5-Large and KGPT (Chen et al., 2020b). Experiment results on Table 9 show that UniD2T exceeds the best competitor KGPT by 1.9% on BLEU and 2.2% on PARENT.

6.4 Further Analysis

6.4.1 Ablation Study

We conduct experiments to investigate the impact of pre-training with graph structure and linear structure. The ablation results are summarized in Table 10, which is divided into two parts: The first part shows the results of directly fine-tuning the pre-trained language model (i.e., T5-Large) on the downstream datasets, referred to as DownData, while the second part presents the results of incorporating additional pre-training data, denoted as PreData, on top of T5-Large. Through careful analysis, we observe that UniD2T (T5-Large+P_{Graph}+F_{Graph}) consistently outperforms T5-Large+F_{Graph} across all six data-to-text datasets, resulting in a notable improvement in the total score of $+20.6$. In addition, we observe that T5-Large+F_{Graph} outperforms T5-Large+F_{Linear} in terms of the total score by $+6.1$. This result clearly indicates that our method significantly improves the performance of the data-to-text models which linearize the structured data as input during fine-tuning the models on downstream datasets. Finally, we delve into the effects of the pre-training datasets. By comparing the results of $P^{*}_{\text{Graph}} + F_{\text{Graph}}$ and $P_{\text{Graph}} + F_{\text{Graph}}$, $P^{*}_{\text{Linear}} + F_{\text{Linear}}$ and $P_{\text{Linear}} + F_{\text{Linear}}$, we observe that the downstream datasets contribute to improving the model's performance and accelerating the pre-training process. It is noteworthy that the pre-training involving both PreData and DownData achieves the best performance across all the experimental datasets.

We also delve into the effects of two Transformer modifications (position and attention matrix construction). The results are illustrated in Table 11. From the results, we observe a significant performance drop when either the structure-aware position or attention matrices are removed, demonstrating the benefits of two Transformer modifications. It is no surprise that combining all the factors achieves the best performance. These findings collectively demonstrate the effectiveness of our proposed method, which explicitly models its graph structure through the use of structure-aware position and attention matrices.

6.4.2 Few-Shot Results

We conduct few-shot experiments on the E2ENLG (Dušek et al., 2020) dataset sourced from the restaurant domain other than Wikipedia. This serves as an additional validation of the model's generalization capabilities. The E2ENLG dataset, assembled through the CrowdFlower platform, encompasses details about restaurants and comprises over 50,000 combinations of dialogue-act-based meaning representations (MRs) with an average of 8.1 references. We fine-tune UniD2T using varying proportions of the training instances

Model	ToTTo	CoSQL	DART	WebNLG	WikiBio	WikiTableT	Total Score
			Only Fine-tuning				
Previous SOTA	49.2	25.6	51.9	57.6	48.6	31.8	–
T5-Large +F_{Linear}	48.1	25.2	50.6	51.7	48.6	31.4	255.6
T5-Large +F_{Graph}	49.1	26.7	51.2	53.1	49.4	32.2	261.7
			With Additional Pre-training				
T5-Large + $P_{\text{Graph}} + F_{\text{Graph}}$ (UniD2T)	**50.2**	**32.7**	**54.9**	**60.4**	**50.4**	**33.7**	**282.3**
T5-Large + $P^*_{\text{Graph}} + F_{\text{Graph}}$	49.3	27.9	53.6	54.7	50.1	32.4	268.0
T5-Large +$P_{\text{Linear}} + F_{\text{Linear}}$	48.7	25.8	53.1	56.7	49.1	31.7	265.1
T5-Large + $P^*_{\text{Linear}} + F_{\text{Linear}}$	48.3	25.7	50.9	52.8	48.7	31.5	257.9

Table 10: Ablation test results on six benchmark datasets. P_{Linear} and P_{Graph} represent the models pre-training with linear structure and graph structure, respectively. F_{Linear} and F_{Graph} represent the models fine-tuning with graph structure and linear structure, respectively. P^* stands for pre-training only with PreData; P indicates pre-training with both PreData and DownData.

Models	BLEU	METEOR	chrF++	TER	BLEURT
UniD2T	**60.4**	**44.4**	**73.4**	**34.1**	**0.65**
- attention	58.6	42.7	70.3	37.2	0.64
- position	58.3	42.6	70.2	36.7	0.64
- all	56.7	42.3	69.8	37.8	0.63

Table 11: Ablation test results on WebNLG test set.

Model	0.1%	0.5%	1%	5%
TGen	3.6	27.9	35.2	57.3
Template-GPT-2	22.5	47.8	53.3	59.9
KGPT-Graph	39.8	53.3	55.1	61.5
KGPT-Seq	40.2	53.0	54.1	61.1
UniD2T	**45.6**	**57.3**	**57.6**	**64.8**

Table 12: Few-shot results on the E2ENLG test set.

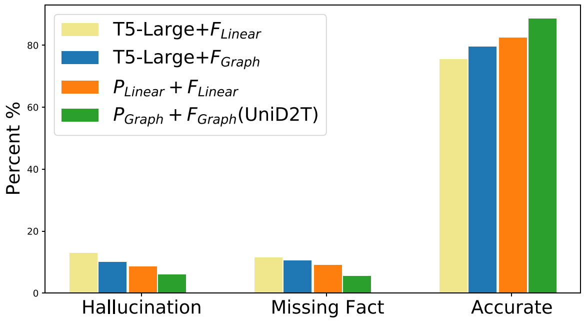


Figure 6: Human evaluation of the factual consistency of different models on WebNLG samples.

(i.e., 0.1%, 0.5%, 1%, 5%, and 10%) from E2ENLG (Dušek et al., 2020). We compare UniD2T with several few-shot learning methods including TGen (Dušek and Jurčíček, 2016), Template-GPT-2 (Chen et al., 2020a), and KGPT (Chen et al., 2020b). The experimental results are summarized in Table 12. We can see that UniD2T significantly outperforms all baselines in various few-shot settings.

6.4.3 Human Evaluation

We also conduct a human evaluation to analyze the generated sentences following Chen et al. (2020b). It is worth noting that each evaluator is unaware of which model generates the text being evaluated so as to avoid evaluation bias. Specifically, we choose 100 test samples from WebNLG and observe the factual consistency between the gold sentences and generated sentences. We invite four NLP workers to assign each text a label from $\{$*Hallucination*, *Missing Fact*, *Accurate*$\}$, similar to (Chen et al., 2020b). As shown in Figure 6, our UniD2T is less prone to hallucinating non-existing facts and can generate more accurate sentences.

6.4.4 Impact on Graph Sizes

To illustrate the effectiveness of the graph structure, we further investigate the performance of $P_{\text{Linear}} + F_{\text{Linear}}$ and $P^*_{\text{Graph}} + F_{\text{Graph}}$ by concerning different graph sizes on the WebNLG validation set. Experimental results in terms of BLEU are shown in Figure 7. When the graph structure is simple, the impact of the graph structure is limited. However, as the graph structure becomes complex, the model with graph structure ($P^*_{\text{Graph}} + F_{\text{Graph}}$) performs much better than the model with linear structure ($P_{\text{Linear}} + F_{\text{Linear}}$). Thus, the structure-enhanced model UniD2T demonstrates

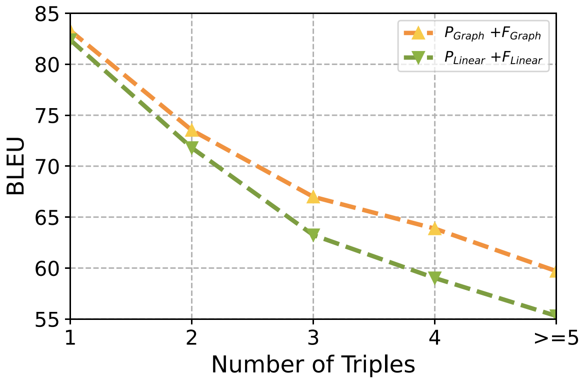


Figure 7: Comparing $P_{Linear} + F_{Linear}$ and $P_{Graph} + F_{Graph}$ BLEU score changes in increasing the number of triples on WebNLG's seen and unseen.

greater stability and better performance on large-scale inputs when compared to linear sequence models.

6.4.5 Impact on Model Sizes

To investigate the influence of different model scales on the experimental results, we conducted experiments using F_{Graph} on T5-Small, T5-Base, T5-Large, and T5-3B on the DART and ToTTo dev sets without pre-training. It is important to note that for our experiments, we conduct evaluations on the dev sets rather than the test sets. This decision is made due to the constraints imposed by the ToTTo dataset, where obtaining test results requires submitting predictions to the leaderboard and awaiting the evaluation process, which can be time-consuming. Therefore, to expedite our research and streamline the experimentation process, we relied on the readily available development sets for conducting our evaluations. The results are presented in the Table 13. Notably, the transition from T5-Large to T5-3B resulted in a substantial increase in the number of parameters by approximately 3.9 times. However, the corresponding improvement in efficacy was found to be less than 1%. This analysis sheds light on the limited impact of scaling up the model size beyond a certain threshold, given the marginal gains in performance despite the significant increase in parameter count.

	ToTTo		DART		
	BLEU	PARENT	BLEU	METEOR	TER
T5-Small + F_{Graph}	45.5	53.3	48.8	0.39	0.45
T5-Base + F_{Graph}	48.6	58.8	50.2	0.40	0.44
T5-Large + F_{Graph}	49.1	59.4	51.2	0.40	0.43
T5-3B + F_{Graph}	49.8	59.7	51.4	0.41	0.43

Table 13: The performance of T5 with different model scales on the dev sets of DART and ToTTo datasets, without performing any pre-training.

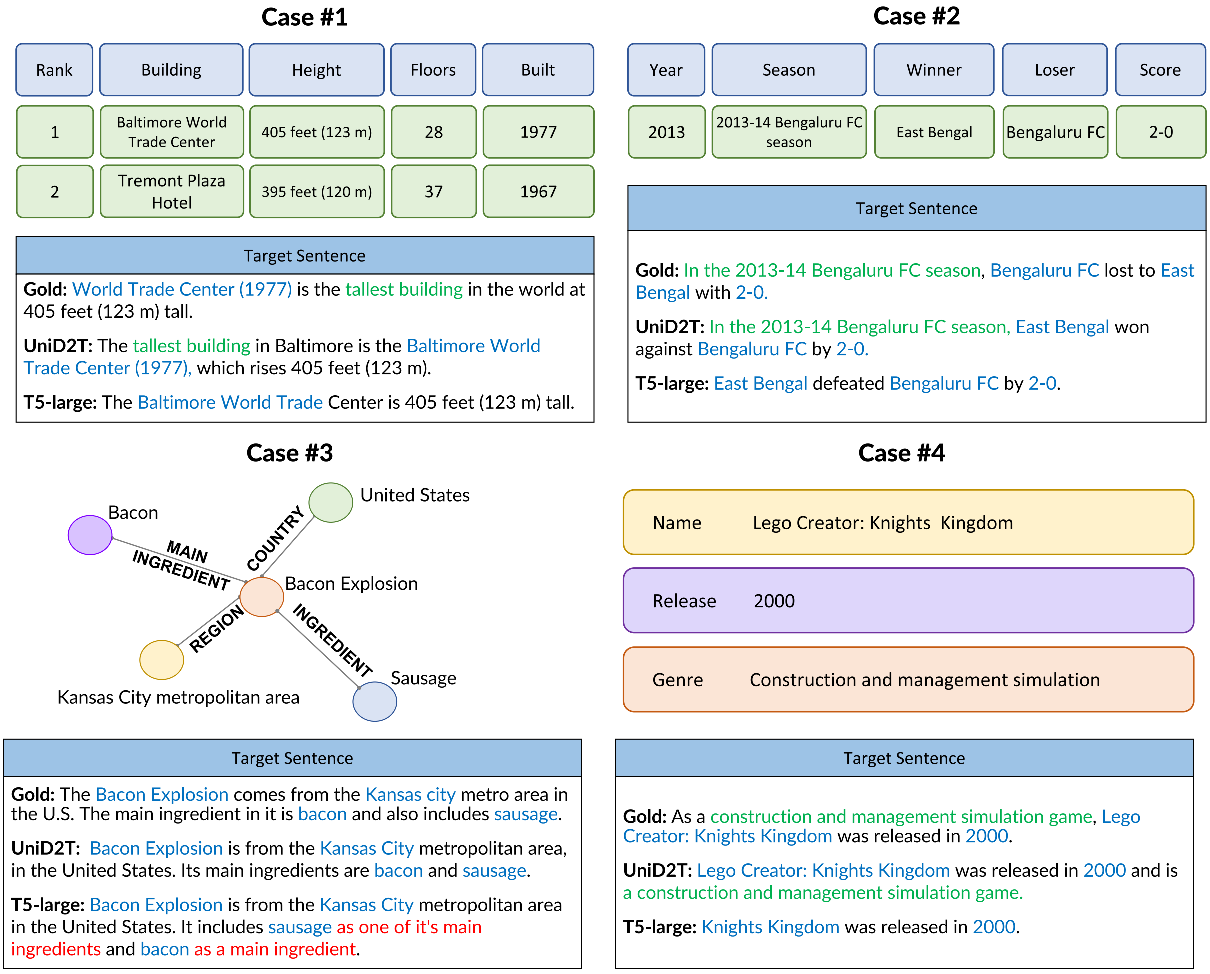
ToTTo
PROMPT: Put the highlighted-table together to form a sentence:
STRUCTURED INPUT: <page_title> List of Malayalam films of 1976 </page_title><table> <cell> Surveykkallu <col_header> Film </col_header> </cell> <cell> Thoppil Bhasi <col_header> Director </col_header> </cell> </table>
DART
PROMPT: Put the triples together to form a sentence:
STRUCTURED INPUT: Mars Hill College: joined: 1973 | Mars Hill College: location: Mars Hill, North Carolina

Table 14: Input examples for ChatGPT on ToTTo and DART. Here, PROMPT represents task description, and STRUCTURED INPUT represents data input with specific formats.

6.5 The Zero-shot Performance of ChatGPT

We conducted zero-shot experiments using ChatGPT on the ToTTo and DART datasets to establish baselines for performance evaluation. The results of these experiments are presented in Table 5 and Table 7 as baselines. The prompt structure of ChatGPT comprises two parts, and detailed information regarding these prompts can be found in Table 14.

From the results, we observe that ChatGPT demonstrates consistent performance across various measures. For instance, in the non-overlap subset of the ToTTo dataset, when compared to BERT-to-BERT, the BLEU score shows a decrease of 17.6%, while the PARENT score exhibits a slight increase of 0.9%. This divergence in BLEU performance indicates that ChatGPT generates responses with different word choices, leading to reduced word overlap with the reference. However, the improvement in the PARENT score suggests enhanced structural and content-related aspects in the generated responses. These findings underscore the importance of employing multiple evaluation metrics to comprehensively assess the



Case #1

Rank	Building	Height	Floors	Built
1	Baltimore World Trade Center	405 feet (123 m)	28	1977
2	Tremont Plaza Hotel	395 feet (120 m)	37	1967

Target Sentence

Gold: World Trade Center (1977) is the tallest building in the world at 405 feet (123 m) tall.

UniD2T: The tallest building in Baltimore is the Baltimore World Trade Center (1977), which rises 405 feet (123 m).

T5-large: The Baltimore World Trade Center is 405 feet (123 m) tall.

Case #2

Year	Season	Winner	Loser	Score
2013	2013-14 Bengaluru FC season	East Bengal	Bengaluru FC	2-0

Target Sentence

Gold: In the 2013-14 Bengaluru FC season, Bengaluru FC lost to East Bengal with 2-0.

UniD2T: In the 2013-14 Bengaluru FC season, East Bengal won against Bengaluru FC by 2-0.

T5-large: East Bengal defeated Bengaluru FC by 2-0.

Case #3

Target Sentence

Gold: The Bacon Explosion comes from the Kansas city metro area in the U.S. The main ingredient in it is bacon and also includes sausage.

UniD2T: Bacon Explosion is from the Kansas City metropolitan area, in the United States. Its main ingredients are bacon and sausage.

T5-large: Bacon Explosion is from the Kansas City metropolitan area in the United States. It includes sausage as one of it's main ingredients and bacon as a main ingredient.

Case #4

Name Lego Creator: Knights Kingdom

Release 2000

Genre Construction and management simulation

Target Sentence

Gold: As a construction and management simulation game, Lego Creator: Knights Kingdom was released in 2000.

UniD2T: Lego Creator: Knights Kingdom was released in 2000 and is a construction and management simulation game.

T5-large: Knights Kingdom was released in 2000.

Figure 8: Examples of generated sentences. The main entity is highlighted in green, and the words that are not faithful to the input are in red. Important information common to both models is indicated in blue.

performance of sophisticated language generation systems in future work.

6.5.1 Impact on Edge Directionality

We take an examination into the significance of edge directionality and present the experimental results of incorporating the edge direction in Table 16. For UniD2T$_{\text{directed}}$, we consider the input directed graph using only its original directed edges (uni-directional) and remove the reverse edges added by UniD2T. Please refer to Section 3.3 for more details about the reverse edges. From Table 16, we can observe that the incorporation of edge direction has a deleterious effect on the performance of pre-trained models. There are several possible factors that may underlie these observed outcomes. (1) First, the pre-training models aim to learn the general representations of structured data. However, due to the vast scale of multi-source data, it is often unfeasible to assign a direction to each data pair. For example, the tabular format constitutes a fundamental type of structured data; however, the absence of explicit edge directionality is a typical characteristic between individual data pairs within this format. Therefore, we default to using bidirectional edges to signify mutual relation ships between two entities. (2) Second, we anticipate learning the coarse relationships between two entities through undirected graphs during the pre-training phase offer greater flexibility to accommodate various types of relationships in different fields. For instance, the directional link ‘‘*Jay Chou* → *Common Jasmine Orange*’’ conveys that *Jay Chou* released the album *Common Jasmine Orange*, while the reverse link ‘‘*Common Jasmine Orange* → *Jay Chou*’’ signifies that *Common Jasmine Orange* is one of *Jay Chou*’s albums. In most cases, it is unnecessary to provide elaborate descriptions of specific relationships, as the data primarily requires indicating connections.

Models	Distinct-1	Distinct-2	Distinct-3	Distinct-4
ChatGPT	**7.56**	**18.93**	**28.33**	**35.75**
T5-Large	6.94	13.94	19.00	23.00
UniD2T	6.58	14.72	21.22	26.38

Table 15: The results of diversity evaluation on DART test set.

6.6 Case Study

As illustrated in Figure 8, we further verify the effectiveness of UniD2T qualitatively by demonstrating some generated sentences by UniD2T and T5-Large. Both UniD2T and T5-Large are capable of generating main entities. However, there are notable differences in the quality and coherence of the generated sentences. Specifically, the sentences generated by T5-Large tend to exhibit shortcomings in terms of including key information and logical reasoning. For instance, in the first case, T5-Large fails to infer that the ''Baltimore World Trade Center'' is the tallest building. This illustrates the limitation of T5-Large in capturing and incorporating specific facts with logical reasoning. In contrast, UniD2T can produce sentences that are more accurate, complete, and encompass the main entities and logical information with greater precision. This highlights the advantages of UniD2T in generating more contextually appropriate and logically grounded sentences.

6.7 The Diversity of Generated Sentences

We conduct an evaluation of the diversity exhibited in the target sentences generated by UniD2T and compare it with strong baselines (i.e., T5-Large and ChatGPT). To quantify the diversity of the generated sentences, we utilized the Distinct-N metric (Li et al., 2016), which calculates the number of distinct N-grams divided by the total number of generated tokens. The experimental results are presented in Table 15, providing insights into the diversity performance of the models. By analyzing the results, it is evident that UniD2T achieves notably higher Distinct-2/3/4 scores compared to T5-Large. This suggests that UniD2T generates sentences with a greater variety of unique unigrams and bigrams than T5-Large, indicating a higher level of linguistic diversity in the output. However, ChatGPT achieves better diversity scores than UniD2T. It tends to generate more diverse words which are not included in our vocabulary, although these words may be non-existing content.

Edge	WikiBio		WikiTableT	
	BLEU	PARENT	BLEU	PARENT
UniD2T	50.4	79.8	33.7	50.7
UniD2T$_{\text{directed}}$	48.8	78.5	31.7	48.3

Table 16: The results of our models with undirected graphs (i.e., UniD2T) and directed graphs (denoted as UniD2T$_{\text{directed}}$), respectively.

6.8 Limitations

Based on our empirical observation, we reveal several limitations of this work, which can be divided into three primary categories. (1) Our pre-training data is limited, which only contains two existing pre-training datasets and six downstream datasets. In the future, we would like to collect more D2T datasets so as to construct a large-scale diverse pre-training corpus. (2) In this work, we unify different structured data into the graph format by using a simple and direct method. We will attempt to exploit more advanced strategies to construct graphs from different structured data. (3) This study focuses on modeling the graph structures and incorporating the structural information into Transformer. However, the pre-training objectives can be further improved so as to further improve the representation learning.

7 Conclusion

In this paper, we proposed a unified data-to-text pre-training method, which could be applied to various downstream data-to-text generation tasks. Concretely, we first converted different types of structured data into graph format. Then, we devised a structure-enhanced Transformer to capture graph structures by introducing two new position and attention matrices to replace the position embedding and attention mask in the self-attention of the Transformer. Extensive experiments on six data-to-text benchmark datasets demonstrated that UniD2T achieved substantially better performance than strong baselines by enabling better information sharing and representation learning of data structures across diverse data-to-text datasets.

Acknowledgments

Min Yang was supported by National Key Research and Development Program of China (2022YFF0902100), National Natural Science Foundation of China (62376262), Shenzhen Science and Technology Innovation Program (KQTD20190929172835662), and Shenzhen Basic Research Foundation (JCYJ20210324115614039 and JCYJ20200109113441941). This work was supported by Alibaba Group through Alibaba Innovative Research Program.

References

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2021. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3554–3565. https://doi.org/10.18653/v1/2021.naacl-main.278

Chenxin An, Jiangtao Feng, Kai Lv, Lingpeng Kong, Xipeng Qiu, and Xuanjing Huang. 2022. Cont: Contrastive neural text generation. *arXiv preprint arXiv:2205.14690*.

Ewa Andrejczuk, Julian Martin Eisenschlos, Francesco Piccinno, Syrine Krichene, and Yasemin Altun. 2022. Table-to-text generation and pre-training with tabt5. *arXiv preprint arXiv:2210.09162*. https://doi.org/10.18653/v1/2022.findings-emnlp.503

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. Graph pre-training for amr parsing and generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6001–6015.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72.

Bin Bi, Chenliang Li, Chen Wu, Ming Yan, Wei Wang, Songfang Huang, Fei Huang, and Luo Si. 2020. Palm: Pre-training an autoencoding&autoregressive language model for context-conditioned generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8681–8691. https://doi.org/10.18653/v1/2020.emnlp-main.700

Deng Cai and Wai Lam. 2020. Graph transformer for graph-to-sequence learning. In *AAAI*. https://doi.org/10.1609/aaai.v34i05.6243

David L. Chen and Raymond J. Mooney. 2008. Learning to sportscast: A test of grounded language acquisition. In *Proceedings of the 25th International Conference on Machine Learning*, pages 128–135. https://doi.org/10.1145/1390156.1390173

Mingda Chen, Sam Wiseman, and Kevin Gimpel. 2021. Wikitablet: A large-scale data-to-text dataset for generating Wikipedia article sections. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 193–209. https://doi.org/10.18653/v1/2021.findings-acl.17

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020a. Logical natural language generation from open-domain tables. *arXiv preprint arXiv:2004.10404*. https://doi.org/10.18653/v1/2020.acl-main.708

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. https://doi.org/10.18653/v1/2020.emnlp-main.697

Jordan Clive, Kris Cao, and Marek Rei. 2021. Control prefixes for text generation. *arXiv preprint arXiv:2110.08329*.

Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William W. Cohen. 2019. Handling divergent reference texts when evaluating table-to-text generation. In *ACL*. https://doi.org/10.18653/v1/P19-1483

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8–14, 2019, Vancouver, BC, Canada*, pages 13042–13054.

Ondřej Dušek and Filip Jurčíček. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. *arXiv preprint arXiv:1606.05491*. https://doi.org/10.18653/v1/P16-2008

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the state-of-the-art of end-to-end natural language generation: The E2E NLG challenge. *Computer Speech & Language*, 59:123–156. https://doi.org/10.1016/j.csl.2019.06.009

Shineng Fang, Jiangjie Chen, Xinyao Shen, Yunwen Chen, and Yanghua Xiao. 2022a. A faithful contrastive framework for response generation in tableqa systems. In *International Conference on Database Systems for Advanced Applications*, pages 197–212. Springer. https://doi.org/10.1007/978-3-031-00129-1_13

Shineng Fang, Jiangjie Chen, Xinyao Shen, Yunwen Chen, and Yanghua Xiao. 2022b. Falcon: A faithful contrastive framework for response generation in tableqa systems. In *International Conference on Database Systems for Advanced Applications*, pages 197–212. Springer. https://doi.org/10.1007/978-3-031-00129-1_13

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for nlg micro-planning. In *55th Annual Meeting of the Association for Computational Linguistics (ACL)*. https://doi.org/10.18653/v1/P17-1017

Zhijiang Guo, Yan Zhang, Zhiyang Teng, and Wei Lu. 2019. Densely connected graph convolutional networks for graph-to-sequence learning. *Transactions of the Association for Computational Linguistics*, 7. https://doi.org/10.1162/tacl_a_00269

Jiuzhou Han and Ehsan Shareghi. 2022. Self-supervised graph masking pre-training for graph-to-text generation. In *Empirical Methods in Natural Language Processing 2022*, pages 4845–4853. Association for Computational Linguistics (ACL). https://doi.org/10.18653/v1/2022.emnlp-main.321

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Mueller, Francesco Piccinno, and Julian Eisenschlos. 2020. Tapas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333. https://doi.org/10.18653/v1/2020.acl-main.398

Mihir Kale and Abhinav Rastogi. 2020. Text-to-text pre-training for data-to-text tasks. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 97–102, Dublin, Ireland. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.inlg-1.14

Pei Ke, Haozhe Ji, Yu Ran, Xin Cui, Liwei Wang, Linfeng Song, Xiaoyan Zhu, and Minlie Huang. 2021. Jointgt: Graph-text joint representation learning for text generation from knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2526–2538. https://doi.org/10.18653/v1/2021.findings-acl.223

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text generation from knowledge graphs with graph transformers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2284–2293.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and

Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

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. https://doi.org/10.18653/v1/D16-1128

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12–17, 2016*, pages 110–119. The Association for Computational Linguistics.

Liang Li, Ruiying Geng, Chengyang Fang, Bing Li, Can Ma, Rongyu Cao, Binhua Li, Fei Huang, and Yongbin Li. 2023a. CATS: A pragmatic Chinese answer-to-sequence dataset with large scale and high quality. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2983–3000, Toronto, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-long.168

Liang Li, Ruiying Geng, Chengyang Fang, Bing Li, Can Ma, Rongyu Cao, Binhua Li, Fei Huang, and Yongbin Li. 2023b. Cats: A pragmatic chinese answer-to-sequence dataset with large scale and high quality. *arXiv preprint arXiv:2306.11477*.

Liang Li, Ruiying Geng, Bowen Li, Can Ma, Yinliang Yue, Binhua Li, and Yongbin Li. 2022. Graph-to-text generation with dynamic structure pruning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6115–6127.

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*, pages 91–99. https://doi.org/10.3115/1687878.1687893

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.

Ao Liu, Haoyu Dong, Naoaki Okazaki, Shi Han, and Dongmei Zhang. 2022. Plog: Table-to-logic pretraining for logical table-to-text generation. *arXiv preprint arXiv:2205.12697*. https://doi.org/10.18653/v1/2022.emnlp-main.373

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. https://doi.org/10.1609/aaai.v32i1.11925

Sanket Vaibhav Mehta, Jinfeng Rao, Yi Tay, Mihir Kale, Ankur Parikh, and Emma Strubell. 2022. Improving compositional generalization with self-training for data-to-text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4205–4219.

Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Mutethia Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chern Tan, Xi Victoria Lin, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani, 2020. Dart: Open-domain structured data record to text generation. *arXiv preprint arXiv:2007.02871*.

Toru Nishino, Ryota Ozaki, Yohei Momoki, Tomoki Taniguchi, Ryuji Kano, Norihisa Nakano, Yuki Tagawa, Motoki Taniguchi, Tomoko Ohkuma, and Keigo Nakamura. 2020. Reinforcement learning with imbalanced dataset for data-to-text medical report generation. In

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 2223–2236. https://doi.org/10.18653/v1/2020.findings-emnlp.202

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318. https://doi.org/10.3115/1073083.1073135

Ankur P. Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. Totto: A controlled table-to-text generation dataset. *arXiv preprint arXiv:2004.14373*. https://doi.org/10.18653/v1/2020.emnlp-main.89

Martin Schmitt, Leonardo F. R. Ribeiro, Philipp Dufter, Iryna Gurevych, and Hinrich Schütze. 2021. Modeling graph structure via relative position for text generation from knowledge graphs. *NAACL-HLT 2021*, page 10. https://doi.org/10.18653/v1/2021.textgraphs-1.2

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395. https://doi.org/10.18653/v1/W15-3049

Ratish Puduppully, Yao Fu, and Mirella Lapata. 2022. Data-to-text generation with variational sequential planning. *Transactions of the Association for Computational Linguistics*, 10:697–715. https://doi.org/10.1162/tacl_a_00484

Alec Radford and Karthik Narasimhan. 2018. *Improving Language Understanding by Generative Pre-Training*. https://api.semanticscholar.org/CorpusID:49313245

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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. https://doi.org/10.1007/978-3-030-45439-5_5

Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87. https://doi.org/10.1017/S1351324997001502

Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2020. Investigating pretrained language models for graph-to-text generation. *arXiv preprint arXiv:2007.08426*. https://doi.org/10.18653/v1/2021.nlp4convai-1.20

Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2021a. Investigating pretrained language models for graph-to-text generation. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 211–227. https://doi.org/10.18653/v1/2021.nlp4convai-1.20

Leonardo F. R. Ribeiro, Yue Zhang, and Iryna Gurevych. 2021b. Structural adapters in pretrained language models for amr-to-text generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4269–4282. https://doi.org/10.18653/v1/2021.emnlp-main.351

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280. https://doi.org/10.1162/tacl_a_00313

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*. https://doi.org/10.18653/v1/2020.acl-main.704

Yunzhou Shi, Zhiling Luo, Pengcheng Zhu, Feng Ji, Wei Zhou, Haiqing Chen, and Yujiu Yang. 2020. G2t: Generating fluent descriptions for knowledge graph. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1861–1864. https://doi.org/10.1145/3397271.3401289

Mathew Snover, Bonnie Dorr, Richard Schwartz, John Makhoul, Linnea Micciulla, and Ralph Weischedel. 2005. A study of translation error rate with targeted human annotation, Technical Report LAMP-TR-126, CS-TR-4755, UMIACS-TR-2005-58.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for amr-to-text generation. In *ACL*. https://doi.org/10.18653/v1/P18-1150

Yixuan Su, David Vandyke, Sihui Wang, Yimai Fang, and Nigel Collier. 2021. Plan-then-generate: Controlled data-to-text generation via planning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 895–909.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8–13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Tianyi Tang, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Mvp: Multi-task supervised pre-training for natural language generation. *arXiv preprint arXiv:2206.12131*. https://doi.org/10.18653/v1/2023.findings-acl.558

Fei Wang, Zhewei Xu, Pedro Szekely, and Muhao Chen. 2022. Robust (controlled) table-to-text generation with structure-aware equivariance learning. *arXiv preprint arXiv:2205.03972*. https://doi.org/10.18653/v1/2022.naacl-main.371

Peng Wang, Junyang Lin, An Yang, Chang Zhou, Yichang Zhang, Jingren Zhou, and Hongxia Yang. 2021. Sketch and refine: Towards faithful and informative table-to-text generation. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1–6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 4831–4843. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.findings-acl.427

Tianming Wang, Xiaojun Wan, and Hanqi Jin. 2020. Amr-to-text generation with graph transformer. *Transactions of the Association for Computational Linguistics*, 8:19–33. https://doi.org/10.1162/tacl_a_00297

Xinyu Xing and Xiaojun Wan. 2021. Structure-aware pre-training for table-to-text generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2273–2278. https://doi.org/10.18653/v1/2021.findings-acl.200

Tao Yu, Rui Zhang, Heyang Er, Suyi Li, Eric Xue, Bo Pang, Xi Victoria Lin, Yi Chern Tan, Tianze Shi, Zihan Li, Youxuan Jiang, Michihiro Yasunaga, Sungrok Shim, Tao Chen, Alexander Fabbri, Zifan Li, Luyao Chen, Yuwen Zhang, Shreya Dixit, Vincent Zhang, Caiming Xiong, Richard Socher, Walter Lasecki, and Dragomir Radev. 2019. CoSQL: A conversational text-to-SQL challenge towards cross-domain natural language interfaces to databases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1962–1979, Hong Kong, China. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1204

Giulio Zhou and Gerasimos Lampouras. 2020. WebNLG challenge 2020: Language agnostic delexicalisation for multilingual RDF-to-text generation. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 186–191, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Jie Zhu, Junhui Li, Muhua Zhu, Longhua Qian, Min Zhang, and Guodong Zhou. 2019. Modeling graph structure in transformer for better amr-to-text generation. In *EMNLP-IJCNLP*. https://doi.org/10.18653/v1/D19-1548