Few-Shot Data-to-Text Generation via Unified Representation and Multi-Source Learning

Alexander Hanbo Li, Mingyue Shang, Evangelia Spiliopoulou, Jie Ma
Patrick Ng, Zhiguo Wang, Bonan Min, William Wang
Kathleen McKeown, Vittorio Castelli, Dan Roth, Bing Xiang
AWS AI Labs

{hanboli, myshang, spilieva, jieman, patricng, zhiguow, bonanmin, wyw}@amazon.com
{mckeownk, vittorca, drot, bxiang}@amazon.com

Abstract

We present a novel approach for structured data-to-text generation that addresses the limitations of existing methods that primarily focus on specific types of structured data. Our proposed method aims to improve performance in multi-task training, zero-shot and few-shot scenarios by providing a unified representation that can handle various forms of structured data such as tables, knowledge graph triples, and meaning representations. We demonstrate that our proposed approach can effectively adapt to new structured forms, and can improve performance in comparison to current methods. For example, our method resulted in a 66% improvement in zero-shot BLEU scores when transferring models trained on table inputs to a knowledge graph dataset. Our proposed method is an important step towards a more general data-to-text generation framework.

1 Introduction

Data-to-text generation is the task of converting structured data into natural language text that can be easily understood by humans. Previous methods for data-to-text generation have been limited to specific structured forms. For example, graph neural networks (GNNs) have been used to encode knowledge graph input (Koncel-Kedziorski et al., 2019; Ribeiro et al., 2020; Guo et al., 2020; Li et al., 2021), while table-specific encoders have been proposed for tables (Liu et al., 2017; Bao et al., 2018; Nema et al., 2018; Jain et al., 2018; Wang et al., 2022). However, these methods are not easily transferable to other structured forms, creating a barrier for scientific development and preventing models from learning across tasks. Recent work has attempted to address the problem of limited structured form applicability by using pretrained language models (PLMs) as a single text-to-text framework for all data structures, by linearizing the data as text sequences. As shown by Kale and Rastogi (2020); Xie et al. (2022), these methods achieve state-of-the-art performance on a wide range of data-to-text tasks.

Despite the advancements made in the field, there are still unresolved questions regarding the relationship between various structured forms, particularly in the context of zero-shot or few-shot settings, where models are required to rapidly adapt to new structured forms. This is particularly pertinent in cases of data scarcity, when structured forms vary across different domains and there is a limited amount of data available for a specific structured form, but a single model is needed to operate on all of them. Such an example is to adapt a knowledge-graph-to-text model to a new domain with data in table format. Even when there is an abundance of data, developing a universal model that can handle all structured forms remains a challenging task. As seen in Xie et al. (2022), a multi-task trained model may perform worse than a single-task model on table inputs. One important reason for such performance drop is because previous research has not fully examined the impact of various linearization methods on these tasks and their effect on cross-task generalization. Despite the use of text-to-text transformers, linearization methods for various structured forms remain diverse, and even within one structured form, linearization can vary across studies. For example, the linearization of KG triples differs in Nan et al. (2021) and Xie et al. (2022), highlighting the need for further research on the relationship between data formats and data-to-text tasks.

In this paper, we address the unresolved questions surrounding the relationship between various structured forms by introducing a *unified representation* for knowledge graphs, tables, and meaning representations. We demonstrate that our method allows for the conversion of knowledge graph triples and meaning representations into virtual tables, which can then be linearized in a consistent manner. Through evaluating our approach on five

representative data-to-text tasks across the afore-mentioned formats, we show that our method not only achieves competitive performance compared to other data-specific linearizations for individual tasks, but also leads to significant improvements in transfer learning scenarios across structured forms, particularly in zero-shot or few-shot settings. For example, using the unified representation improves the zero-shot BLEU score by relatively 66% when transferring from ToTTo (Parikh et al., 2020) to DART (Nan et al., 2021). Additionally, our approach results in improved performance when used in multi-task settings compared to models trained with varied linearizations. These results provide a clear indication of the effectiveness of our proposed unified representation in enhancing cross-task generalization.

2 Related Work

Data-Type Specific Knowledge Encoding Research has been conducted to encode structured knowledge using various models and approaches, including Graph Neural Networks (GNNs) (Koncel-Kedziorski et al., 2019; Ribeiro et al., 2020; Guo et al., 2020; Li et al., 2021; Song et al., 2018; Ribeiro et al., 2019; Cai and Lam, 2020; Zhang et al., 2020; Ribeiro et al., 2021b; Schmitt et al., 2021) and neural encoder-decoder models based on Gated Recurrent Units (GRUs) and Transformers (Gehrmann et al., 2018; Ferreira et al., 2019). These models have been used to assist in encoding knowledge graph inputs and meaning representations. Additionally, several models have been proposed for table-to-text generation, including approaches that combine content selection or entity memory in a Long Short-Term Memory (LSTM) model (Puduppully et al., 2018, 2019), and others that focus on table-specific encoders (Liu et al., 2017; Bao et al., 2018; Nema et al., 2018; Jain et al., 2018). More recent studies have utilized the capabilities of pre-trained language models in their designs, but have also incorporated specialized encoder structures or attention mechanisms specifically for table inputs. These include encoder-only models (Arik and Pfister, 2019; Yin et al., 2020; Herzig et al., 2020; Huang et al., 2020; Wang et al., 2021; Iida et al., 2021; Eisenschlos et al., 2021; Yang et al., 2022), as well as encoder-decoder models (Cao, 2020; Andrejczuk et al., 2022; Wang et al., 2022). However, it should be noted that the encoder structures of these works are specifically tailored

for table input and cannot be directly applied to other types of data.

Structured Data Linearization Recent developments in pretrained language models (Devlin et al., 2019; Radford et al., 2019; Lewis et al., 2020; Raffel et al., 2020) have made it possible to use a single text-to-text framework for various types of data by linearizing them as text sequences. Studies have been conducted on finetuning PLMs on table input (Parikh et al., 2020) and knowledge graph input (Kasner and Dušek, 2020; Ribeiro et al., 2021a), single-task and multi-task training on a collection of structured data grounding tasks (Xie et al., 2022), and the effectiveness of pretraining and fine-tuning strategies for data-to-text tasks (Kale and Rastogi, 2020) and table-based question answering tasks (Shi et al., 2022). These studies have consistently found that linearizing structured data as a sequence of tokens without modifying the model structure, is a simple yet effective strategy that outperforms pipelined neural architectures specifically tailored to particular data types.

Zero/Few-Shot Data-to-Text Generation The studies such as Chen et al. (2020b) and Ke et al. (2021) have evaluated the zero and few-shot performance of PLMs on knowledge graph input, highlighting the benefits of a joint pretraining strategy on knowledge graphs and texts for learning better KG representations. Keymanesh et al. (2022) studied the prompt-tuning method for KG-to-text generation and found it to be effective in a few-shot setting. Chen et al. (2020d) combines PLM with a table content selector using a switch policy. Other researchers have also explored methods such as data augmentation (Chang et al., 2021) and retrieval-based input augmentation (Su et al., 2021) to aid in few-shot data-to-text generation. Kasner and Dusek (2022) proposes a pipeline approach involving a sequence of operations, such as ordering and aggregation, and only finetunes the PLMs of these modules to make the pipeline more domain-independent.

3 Unified Representation

In this section, we demonstrate that structured data, such as tables, highlighted cells, knowledge graph triples, and meaning representations, can be linearized in a consistent manner. We begin by showing in Section 3.1 how knowledge graph triples and meaning representations can be mapped to a virtual
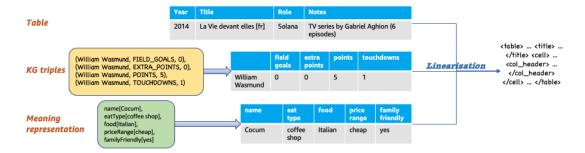
Figure 1: unified representation of three data types: table, KG triples, and meaning representations. The latter two are first converted to virtual tables, and then linearized using the same method as table input.

table and subsequently linearized in the same way as tables. Next, in Section 3.2, we demonstrate the process of linearizing a table or highlighted cells. The entire method is illustrated in Figure 1.

3.1 Virtual Table

KG Triple The method for converting triples from a connected sub-graph into a virtual table involves using the tail node of each triple as a cell value and the relation as the column header. Nodes that do not appear as tail nodes are not assigned a column header. An example is provided in Figure 1. "William Wasmund" does not have a column header assigned since it never appears as a tail node. If a set of knowledge graph triples contains multiple connected components, each component is converted into a separate table.

Meaning Representation We focus on textual MRs that appear as a list of comma-separated attribute-value pairs (Dušek et al., 2020). These MRs can be treated as virtual tables by associating each Attribute[Value] with a cell value, represented by the "Value", and the "Attribute" as its corresponding column header. An example of this can be seen in Figure 1.

3.2 Linearization of Tables

After converting both KGs and MRs into virtual tables, we end up with only table inputs that need to be linearized. In this section, we discuss one choice of such a linearization method, motivated by ToTTo linearization (Parikh et al., 2020). Additionally, we will provide a specific example of how to linearize Table 1 in the following sections.

Basic Units The basic units for linearization are presented in Table 2. Each unit is defined by a start symbol, <xx>, and an end symbol, </xx>.

Table Title: Alma Jodorowsky
Section Title: Filmography

Year	Title	Role
2014	La Vie devant elles [fr]	Solana
2016	Kids in Love	Evelyn
2017	The Starry Sky Above Me	Justyna

Table 1: An example table to showcase our linearization.

Start Symbol	Meaning
<table>	contents in a table
<column>	contents in a column
<row>	contents in a row
<cell>	content in a cell
<col_header>	column header name
<row_header>	row header name
<title>	main title /domain / topic of the input
<sub_title>	sub-title /domain /topic of the input

Table 2: Basic units of our linearization.

Linearization of Highlighted Cells To linearize the highlighted cells, we proceed in a left-to-right, top-to-bottom order. For instance, in Table 1, the linearization of the highlighted cells (in yellow background) appears as follows: [1]

```
1  <title> Alma Jodorowsky </title>
2  <sub_title> Filmography </sub_title>
3  <table>
4    <cell> 2016
5      <col_header> Year </col_header>
6    </cell>
7    <cell> Kids in Love
8      <col_header> Title </col_header>
9    </cell>
10   <cell> Evelyn
11     <col_header> Role </col_header>
12   </cell>
13 </table>
```

---

[1]Indentation is used for clarity in this example, but it is not present in the actual input.

Linearization of (Sub)Table A row-wise linearization of the entire Table 1 is:

```
1  <title> Alma Jodorowsky </title>
2  <sub\_title> Filmography </sub\_title>
3  <table>
4    <row>
5      <cell> 2014
6        <col_header> Year </col_header>
7      </cell>
8      <cell> La Vie devant elles [fr]
9        <col_header> Title </col_header>
10     </cell>
11     <cell> Solana
12       <col_header> Role </col_header>
13     </cell>
14   </row>
15   ...(other rows)...
16 </table>
```

Such a linearization method can also be applied to column-wise. An example is provided in the Appendix B.

4 Experiments

Datasets We test our method on five data-to-text datasets: The **ToTTo** dataset (Parikh et al., 2020) poses the challenge of generating a one-sentence description, given highlighted cells from a Wikipedia table. Our models are evaluated on the validation set, as the annotations for the test set are not publicly available. The **DART** corpus (Nan et al., 2021) is an open-domain structured data-to-text resource, consisting of entity-relation triples. The **LogicNLG** dataset (Chen et al., 2020a) investigates the ability to generate logical inferences from table contents to implicit insights, as the target sentences. The **WebNLG** dataset (Gardent et al., 2017) includes triples from 15 DBpedia categories, which are mapped to their verbalization. Results are reported on the Seen (S), Unseen (U), and All (A) subsets of the data. The **E2E clean** dataset (Dušek et al., 2019) consists of meaning representations (MRs) from the restaurant domain. The task is to generate a sentence that verbalizes the *useful* information from the MR. Dataset statistics are summarized in Table 7 in the appendix.

Evaluation Metrics We evaluate the quality of generated texts using several widely accepted metrics. *BLEU* (Papineni et al., 2002) measures the similarity between generated text and references in terms of n-gram overlap. *METEOR* (Banerjee and Lavie, 2005) assesses the quality of generated text by comparing unigram matches between the text and references, including exact, stem, synonym, and paraphrase matches. *TER* (Snover et al., 2006) is a measure of the number of edits required to

change the generated text into one of the references. *PARENT* (Dhingra et al., 2019) takes into account the table input when evaluating generated text. *NIST* (Doddington, 2002) is similar to BLEU, but also considers the informativeness of each n-gram. *CIDEr* (Vedantam et al., 2015) uses TF-IDF to lower the weights of common n-grams that appear in all references when calculating uni-gram to 4-gram overlaps between generated and reference sentences. We also use the *NLI score* (Chen et al., 2020a) on the LogicNLG dataset to evaluate the logical fidelity, which is a model-based evaluation using the BERT model trained on the TabFact (Chen et al., 2020c) dataset.

Comparing Linearizations We compare our proposed *unified representation* to other linearization methods from previous papers. Specifically, on DART, WebNLG, and E2E datasets, we compare our method to the linearization used in Unified-SKG (Xie et al., 2022).[2] On ToTTo and LogicNLG datasets, we use the linearization from their original papers (Parikh et al., 2020; Chen et al., 2020a) for comparison. Examples of their linearization methods can be found in the appendix.

4.1 Zero and Few-Shot Experiments

Our hypothesis is that a model trained on one structured form will transfer better to other forms under zero or few-shot settings when using our unified method of representation. We test this by focusing on transferring from ToTTo data (table input) to other types and from WebNLG (KGs) to ToTTo in this section. Results for other transfers can be found in the appendix.

Setting	Src representation	Tgt representation
Only on tgt	-	Others
Src to tgt, unified	Unified	Unified
Src to tgt, varied	Others	Others

Table 3: Comparison of source and target task representations. "Unified" uses our proposed unified representation, "Others" uses linearizations from other papers for each task.

As shown in Table 3, for each experiment, we compare **three settings**: (i) *Only on tgt* – In few-shot experiments, we only train the model on the target task using the linearization from other papers. In zero-shot experiments, we use the foundational

---

[2]The E2E dataset is not studied in the paper, but the linearization is included in their official repository.

model without any training. (ii) *Src to tgt, unified* –
First, train the model on the source task and then
fine-tune it on k-shot[3] target-task data, using our
unified representation for both. (iii) *Src to tgt, var-
ied* – Similar to (ii), but we use the linearization
from other papers for each task, as described in 4.
We refer to this as the varied setting because the
source and target-task linearizations are different.

During inference, we apply the same lineariza-
tion method utilized during training to each target
task. More implementation details are presented in
the appendix.

4.1.1 Zero-Shot Performance

The zero-shot results are summarized in Table 4.
We compare our results to recent works GPT2-
XL (Keymanesh et al., 2022), KGPT (Chen et al.,
2020b), JointGT (Ke et al., 2021) and HTLM
(Aghajanyan et al., 2022). Both KGPT and JointGT
models are pretrained on large amounts of aligned
knowledge graph and text data. HTLM is a hyper-
text language model pre-trained on a large-scale
web crawl. It allows for structured prompting in
the HTML format.

From the results, we make several observations.
(1) The *Only on tgt* performance is very low as ex-
pected, as the T5-base model has not been trained
on any data. However, surprisingly the NLI score
on LogicNLG is the highest under this setting. We
observe that this NLI score is very unstable and
might not be a good metric for judging the entail-
ment of generated text. **(2)** The performance of
Src to tgt, unified consistently and significantly sur-
passes that of *Src to tgt, varied*, even though both
models are trained using the same source-task data,
but with different representations. This demon-
strates that representing source and target tasks in
the same format is crucial for successful zero-shot
transfer, as a common representation facilitates the
transfer of knowledge learned on the source data to
other structured forms and tasks. **(3)** The zero-shot
performance of the "unified" model is even better
than few-shot results of the baseline models. On
DART, the "unified" model's BLEU score is 43%
higher than that of HTLM. The improvement on
WebNLG is particularly noteworthy for unseen cat-
egories. Utilizing a unified representation results
in a zero-shot BLEU score of 39.82, surpassing the
few-shot results of 37.18 by Ke et al. (2021) and
18.5 by Aghajanyan et al. (2022).

---

[3]$k = 0$ means no training on target task at all.

4.1.2 Few-Shot Results

Figure 2 shows the few-shot results for sample sizes
8, 16, 32, 64, and 128. We repeat the experiments
5 times for each sample size and report the mean
and 95% confidence intervals.

Table ⟶ KG Triples   From Figure 2a, 2b and
2c, we have identified three key observations: (1)
Both the models *Src to tgt, unified* and *Src to tgt,
varied*, which were initially trained on ToTTo, per-
form significantly better than the model *Only on
tgt*, which was only trained on target tasks. This
indicates that these two structured forms share com-
mon knowledge and that training the model on tab-
ular input can greatly enhance its understanding
of KG triples. (2) Furthermore, *Src to tgt, unified*
(represented by the red curve) outperforms *Src to
tgt, varied* (represented by the blue curve) by a
substantial margin. This observation aligns with
our previous findings in the zero-shot setting (as
seen in Table 4) and highlights the importance of
our unified representation approach in transferring
knowledge learned from tables to KG triples. (3)
Additionally, on the task of WebNLG, the improve-
ment on unseen categories is particularly notable,
further reinforcing our zero-shot findings.

Table ⟶ Meaning Representations   Based on
Figure 2d, similar observations can be made for
the E2E dataset. The improvement in terms of
CIDEr is particularly significant when using fewer
than 64 samples, indicating that the unified model
generates more informative text compared to the
varied and vanilla models.

Table Description ⟶ Table Insights   The Log-
icNLG task is distinct from the ToTTo task in that
it requires the model to generate insights by ana-
lyzing the contents of a table, rather than generat-
ing surface-form descriptions based on highlighted
cells. As shown in Figure 2e, when using only 8
samples, the *Src to tgt, varied* model performs bet-
ter than the *Src to tgt, unified* model. This may be
due to the fact that both tasks involve generating
text from tables, and that the unified model is more
proficient at transferring knowledge learned on the
source task to the target task, which may lead to the
generation of table descriptions rather than insights
when provided with a limited number of samples.
However, as the number of samples increases, the
performance of the unified model improves, and it
surpasses the varied model when k=128. A con-
crete example is provided in the case study section

Setting	DART (KG)			WebNLG (KG)			E2E clean (MR)			LogicNLG (Table)	
	BLEU	MET	TER↓	S	U	A	BLEU	NIST	CIDEr	BLEU-3	NLI
GPT2-XL	13.3	0.24	0.65	-	-	-	-	-	-	-	-
KGPT	-	-	-	-	-	13.9	-	-	-	-	-
JointGT (0.5%)[a]	-	-	-	-	37.2	-	-	-	-	-	-
HTLM (1-shot)[a]	22.1	0.12	0.91	28.1	18.5	22.8	-	-	-	-	-
Only on tgt[b]	0.3	0.01	2.82	0.36	0.08	0.23	0.0	0.0	0.0	0.2	**85.1**
Src to tgt, varied	18.9	0.21	1.00	34.1	28.5	31.3	12.1	2.8	0.3	7.8	70.9
Src to tgt, unified	**31.5**	**0.32**	**0.56**	**35.9**	**39.8**	**37.7**	**22.6**	**4.4**	**0.9**	**8.9**	81.3

[a] *We compare our results to their few-shot performance, as zero-shot results are not reported in their papers.*

[b] *Under zero-shot, this means directly testing T5-base model on target test set without any training.*

Table 4: Zero-shot results. Our foundational model is T5-base (220M). MET stands for METEOR, and lower scores on TER indicate better performance. On WebNLG, BLEU scores are reported for seen (S), unseen (U), and all (A) categories. The NLI-accuracy is calculated using the NLI model provided in LogicNLG official codebase. On papers without zero-shot results, we report their few-shot performance.

(a) ToTTo (table) to DART (KG): **BLEU**

(b) ToTTo (table) to WebNLG (KG): **BLEU (Unseen)**

(c) ToTTo (table) to WebNLG (KG): **BLEU (Seen)**

(d) ToTTo (table) to E2E (MR): **CIDEr**

(e) ToTTo to LogicNLG (table): **BLEU**

(f) WebNLG (KG) to ToTTo (table): **PARENT**

Figure 2: Results of few-shot experiments transferring models between two structured forms. Each figure shows three curves, the green curve *"only on tgt"* is the performance of the T5-base model fine-tuned directly on the target task, the red curve *"src to tgt, unified"* is the performance of the model fine-tuned on both tasks using our proposed unified representation, and the blue curve *"src to tgt, varied"* is the performance of the model fine-tuned on both tasks using linearization from other papers, resulting in varied linearization for source and target tasks. The LogicNLG task differs from ToTTo by requiring the model to generate insights from analyzing a table rather than generating descriptions from highlighted cells.

4.3 to further illustrate our observation.

KG Triples ⟶ Table The benefits of utilizing unified representation are particularly substantial when transferring models that have been trained on knowledge graphs to table inputs. In Figure 2f, the PARENT gap between unified and varied models is consistently greater than 2 points. In fact, the performance of "varied" and "only on tgt" models converge when utilizing 128 samples, and is only slightly superior to that of the "unified"

model when provided with only 8 samples. This suggests that the use of unified representation is highly efficient in terms of sample utilization.

4.2 Full-Set Finetuning Results

In this section, we train the models on full training sets, in either single-task or multi-task settings. Additional experimental results are presented in the appendix.

Model	Linear	ToTTo BLEU	ToTTo PARENT	DART BLEU	WebNLG S	WebNLG U	WebNLG A	LogicNLG BLEU-3	E2E BLEU	E2E CIDEr
Single-Task Training										
LATTICE(Wang et al., 2022)	Tab	48.6	-	-	-	-	-	20.1	-	-
UnifiedSKG (base)	O	48.3	-	46.2	-	-	-	-	-	-
UnifiedSKG (3B)	O	49.0	-	46.7	-	-	-	-	-	-
DCVED(Chen et al., 2021)	O	-	-	-	-	-	-	15.3	-	-
HTLM(Aghajanyan et al., 2022)	O	-	-	47.2	65.4	48.4	55.6	-	-	-
T5-base	Uni	49.3	58.9	48.6	65.4	50.1	58.5	24.7	41.8	1.90
	O	49.2	58.9	49.0	**65.9**	49.5	58.2	25.2	42.1	1.91
T5-3B	Uni	49.4	58.9	**49.6**	65.1	52.7	59.5	25.1	**42.8**	1.92
	O	**49.6**	**59.0**	49.3	65.3	**53.5**	**60.0**	25.3	42.5	**1.94**
Multi-Task Training										
UnifiedSKG (base)	O	45.3	-	-	-	-	-	-	-	-
C-P (large) (Clive et al., 2021)	O	-	-	52.0	67.0	55.6	61.8	-	44.2	-
T5-base	Uni	49.7	59.2	49.8	64.9	50.3	58.3	25.2	42.9	1.94
	O	48.5	58.7	48.1	64.1	50.2	57.9	24.7	41.7	1.89
T5-3B	Uni	**50.8**	**60.4**	**50.2**	**65.4**	**53.4**	**60.0**	**25.4**	**43.2**	**1.99**
	O	50.2	59.5	49.8	65.3	51.9	59.4	25.3	41.8	1.89

Table 5: Single-task and multi-task training results using full training sets. In the "Linear" column, "Uni" represents using unified representation, "O" means using other linearizations from previous papers, and "Tab" mean we use table-specific encoder.

Single-Task Training From the "single-task training" results in Table 5, a key finding is that the proposed unified representation method results in performance comparable to other linearization techniques studied in previous research. This is particularly evident on the DART, WebNLG, and E2E tasks, where the data was first converted into virtual tables, and the results from both methods are similar, indicating that this conversion does not result in a significant loss of information.

Multi-Task Training The performance of multi-task models is summarized in Table 5 under the "multi-task training" section, revealing several key findings: **(1)** *Overall, multi-task training using different linearizations for each dataset results in a **worse** performance compared to single-task training.* BLEU scores for T5-base models decrease from 49.2 to 48.5 on ToTTo, from 49.0 to 48.1 on DART, and from 65.9 to 64.1 on seen categories of WebNLG. This confirms the findings of Unified-SKG (Xie et al., 2022), which found that single-task model performance was higher than multi-task performance on ToTTo dataset. However, it is unclear if this drop in performance was due to task differences, as their study included other tasks. Our results provide further insight into data-to-text tasks alone and show that multi-task performance can still be inferior if input formats are not unified. **(2)** In contrast, *multi-task trained "unified" models consistently outperform single-task models,*

with the only exception of the base model on the WebNLG dataset. This demonstrates that utilizing a unified representation approach helps models learn common knowledge across various tasks without negatively impacting performance. **(3)** *The "unified" models consistently demonstrate superior performance compared to "varied" models in multi-task training*, with a larger margin of improvement observed in base-sized models.

4.3 Qualitative Study

We conduct a qualitative case study to compare the texts generated by the *Src to tgt, unified* and *Src to tgt, varied* models. The results are illustrated in Table 6, which displays the model's generations for different sample sizes.

For the WebNLG example, the input contains 5 KG triples. When $k = 8$, the "varied" model only covers one KG triple fact, while the "unified" model includes many more nodes and relations from the input. As the sample size increases to 128, the "unified" model's generation covers all facts accurately, while the "varied" model's generation still misses the "funk and disco" origin of pop music.

In the E2E example, the "unified" model output is consistent and accurate with both 8 and 128 samples. In contrast, the "varied" model produces "Sorrento" twice. This serves as additional evidence that using a unified representation enhances the transfer of the generation style learned on table input to meaning representations.

k-shot =	Src to tgt, unified	Src to tgt, varied
	ToTTo (table) ⟶ WebNLG (KG) example	
8	Hip hop music is influenced by Disco by Allen Forrest (born in Fort Campbell) and Funk with drum and bass.	Allen Forrest was born in Fort Campbell.
128	Allen Forrest, born in Fort Campbell, is known for his roots in hip hop music. `Disco and Funk` are stylistic origins, while drum and bass are derivatives.	Allen Forrest was born in Fort Campbell and is known for hip hop music. Hip hop music is a derivative of drum and bass.
Groundtruth	Allen Forrest was born in Fort Campbell and is a hip hop musician. Hip hop originates from `funk and disco` and was derived into drum and bass music.	
KG triples	(Hip hop music, stylistic origin, `Disco`) (Allen Forrest, birth place, Fort Campbell) (Allen Forrest, genre, Hip hop music) (Hip hop music, stylistic origin, `Funk`) (Hip hop music, derivative, Drum and bass)	
	ToTTo (table) ⟶ E2E (MR) example	
8	Zizzi is a pub near The Sorrento.	Zizzi is a gastropub in Sorrento, near The Sorrento.
128	Zizzi is a pub near The Sorrento.	Zizzi is a pub near The Sorrento.
Groundtruth	There is a pub called Zizzi located near The Sorrento.	
MRs	name[Zizzi], eatType[pub], near[The Sorrento]	
	ToTTo (table) ⟶ LogicNLG (table) example	
8	In the world golf championships, the United States has 12 individual winners, Australia has 3 individual winners, England has 3 individual winners, South Africa has 1 individual winner, Canada has 1 individual winner, Fiji has 1 individual winner, Italy has 1 individual winner, Japan has 0 individual winner, and Wales has no individual winner.	The United States has the highest number of individual winners of any country in the world.
128	The United States is the only nation to have won 12 World Golf Championship.	The United States has the highest number of individual winners.

Nation	United States	Australia	England	South Africa	Northern Ireland	Germany	Canada	Fiji	Sweden	Italy	Japan	Wales
Individual winner	12	3	3	1	1	1	1	1	1	1	0	0

Input table

Table 6: Case study: few-shot prediction examples on WebNLG and E2E.

The results of the LogicNLG input generation offer validation for our hypothesis that the "unified" model performs less effectively than the "varied" model when the sample size is small, due to its persistent focus on generating descriptions of the table input, as it has been trained to do on the ToTTo data. Indeed, the descriptions generated by the "unified" model when sample size is 8, are accurate reflections of the table's content. When the sample size is increased to 128, both models generate sentences that are more akin to insights. It is noteworthy that the "unified" model generates "world golf championship" even though it is not present in the table, which pertains to the golf championship. We posit that this information is carried over from the ToTTo data, and the "unified" model is able to retain this information while the "varied" model does not.

5 Conclusion and Future Work

We have introduced a unified representation approach for data-to-text tasks, which effectively converts table contents, knowledge graph triples, and meaning representations into a single representation. Our experiments demonstrate that this unified representation significantly improves generalization across different structured forms, especially in zero-shot or few-shot settings. Our method is particularly beneficial in situations where data is scarce. Additionally, by using the unified representation, our multi-task-trained models consistently outperform single-task models, which is in contrast to previous findings that mixing different data types can negatively impact overall performance.

One future direction is to apply our method to other tasks that involve heterogeneous inputs, such as question answering over knowledge bases, where knowledge can be stored in both tables and knowledge graphs. It would also be interesting to investigate whether a model pre-trained on large knowledge graphs can more effectively transfer learned commonsense knowledge to table QA tasks, when using our unified representation approach.

Limitations

It is important to note that the unified representation proposed in our study is just one option among many. Other linearization methods may potentially yield better results. For example, research by Yin et al. (2022) and Aghajanyan et al. (2022) has explored using code generation with Jupyter notebooks and a hyper-text language model with structured prompting, respectively. Further research in these areas, such as converting all structured forms to markdown language or hyper-texts, may yield alternative unified representations.

Ethics Statement

We acknowledge the importance of the ACL Ethics Policy and agree with it. This study addresses the problem of data-to-text generation and explores whether a unified representation can enhance cross-task performance on various structured forms. Since our input comes from knowledge bases, a potential concern is that biases or fairness issues may be present in the KB, which could also be reflected in the generated text. Therefore, it is crucial to use the model with caution in practice. We believe this work can contribute to the field of data-to-text generation, particularly in situations where data is scarce.

References

Armen Aghajanyan, Dmytro Okhonko, Mike Lewis, Mandar Joshi, Hu Xu, Gargi Ghosh, and Luke Zettlemoyer. 2022. HTLM: Hyper-text pre-training and prompting of language models. In *International Conference on Learning Representations*.

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*.

Sercan Ö. Arik and Tomas Pfister. 2019. Tabnet: Attentive interpretable tabular learning. *ArXiv*, abs/1908.07442.

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, Ann Arbor, Michigan. Association for Computational Linguistics.

Junwei Bao, Duyu Tang, Nan Duan, Zhao Yan, Yuanhua Lv, M. Zhou, and Tiejun Zhao. 2018. Table-to-text: Describing table region with natural language. In *AAAI Conference on Artificial Intelligence*.

Deng Cai and Wai Lam. 2020. Graph transformer for graph-to-sequence learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7464–7471.

Juan Cao. 2020. Generating natural language descriptions from tables. *IEEE Access*, 8:46206–46216.

Ernie Chang, Xiaoyu Shen, Dawei Zhu, Vera Demberg, and Hui Su. 2021. Neural data-to-text generation with LM-based text augmentation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 758–768, Online. Association for Computational Linguistics.

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020a. Logical natural language generation from open-domain tables. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7929–7942, Online. Association for Computational Linguistics.

Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. 2020b. KGPT: Knowledge-grounded pre-training for data-to-text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8635–8648, Online. Association for Computational Linguistics.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020c. Tabfact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*.

Wenqing Chen, Jidong Tian, Yitian Li, Hao He, and Yaohui Jin. 2021. De-confounded variational encoder-decoder for logical table-to-text generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5532–5542, Online. Association for Computational Linguistics.

Zhiyu Chen, Harini Eavani, Wenhu Chen, Yinyin Liu, and William Yang Wang. 2020d. Few-shot NLG with pre-trained language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 183–190, Online. Association for Computational Linguistics.

Jordan Clive, Kris Cao, and Marek Rei. 2021. Control prefixes for text generation. *CoRR*, abs/2110.08329.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William Cohen. 2019. Handling divergent reference texts when evaluating table-to-text generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4884–4895, Florence, Italy. Association for Computational Linguistics.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, page 138–145, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Ondřej Dušek, David M. Howcroft, and Verena Rieser. 2019. Semantic noise matters for neural natural language generation. In *Proc. of the 12th International Conference on Natural Language Generation*, pages 421–426, Tokyo, Japan. Association for Computational Linguistics.

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.

Julian Eisenschlos, Maharshi Gor, Thomas Müller, and William Cohen. 2021. MATE: Multi-view attention for table transformer efficiency. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7606–7619, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thiago Castro Ferreira, Chris van der Lee, Emiel Van Miltenburg, and Emiel Krahmer. 2019. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. 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 552–562.

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

Sebastian Gehrmann, Falcon Dai, Henry Elder, and Alexander M Rush. 2018. End-to-end content and plan selection for data-to-text generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 46–56.

Qipeng Guo, Zhijing Jin, Xipeng Qiu, Weinan Zhang, David Wipf, and Zheng Zhang. 2020. CycleGT: Unsupervised graph-to-text and text-to-graph generation

via cycle training. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 77–88, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, 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, Online. Association for Computational Linguistics.

Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar S. Karnin. 2020. Tabtransformer: Tabular data modeling using contextual embeddings. *ArXiv*, abs/2012.06678.

Hiroshi Iida, Dung Thai, Varun Manjunatha, and Mohit Iyyer. 2021. TABBIE: Pretrained representations of tabular data. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3446–3456, Online. Association for Computational Linguistics.

Parag Jain, Anirban Laha, Karthik Sankaranarayanan, Preksha Nema, Mitesh M. Khapra, and Shreyas Shetty. 2018. A mixed hierarchical attention based encoder-decoder approach for standard table summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 622–627, New Orleans, Louisiana. Association for Computational Linguistics.

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.

Zdeněk Kasner and Ondřej Dušek. 2020. Train hard, finetune easy: Multilingual denoising for RDF-to-text generation. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 171–176, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Zdeněk Kasner and Ondrej Dusek. 2022. Neural pipeline for zero-shot data-to-text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3914–3932, Dublin, Ireland. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Moniba Keymanesh, Adrian Benton, and Mark Dredze. 2022. What makes data-to-text generation hard for pretrained language models? *ArXiv*, abs/2205.11505.

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, Minneapolis, Minnesota. 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.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Zhicheng Wei, Nicholas Jing Yuan, and Ji-Rong Wen. 2021. Few-shot knowledge graph-to-text generation with pretrained language models. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1558–1568, Online. Association for Computational Linguistics.

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2017. Table-to-text generation by structure-aware seq2seq learning. *arXiv preprint arXiv:1711.09724*.

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. 2021. DART: Open-domain structured data record to text generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 432–447, Online. Association for Computational Linguistics.

Preksha Nema, Shreyas Shetty, Parag Jain, Anirban Laha, Karthik Sankaranarayanan, and Mitesh M. Khapra. 2018. Generating descriptions from structured data using a bifocal attention mechanism and gated orthogonalization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1539–1550, New Orleans, Louisiana. Association for Computational Linguistics.

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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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*.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2018. Data-to-text generation with content selection and planning. *arXiv e-prints*, pages arXiv–1809.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with entity modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2023–2035.

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

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.

Leonardo F. R. Ribeiro, Claire Gardent, and Iryna Gurevych. 2019. Enhancing AMR-to-text generation with dual graph representations. 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 3183–3194, Hong Kong, China. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Leonardo F. R. Ribeiro, Yue Zhang, Claire Gardent, and Iryna Gurevych. 2020. Modeling global and local node contexts for text generation from knowledge graphs. *Transactions of the Association for Computational Linguistics*, 8:589–604.

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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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. In *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)*, pages 10–21, Mexico City, Mexico. Association for Computational Linguistics.

Peng Shi, Patrick Ng, Feng Nan, Henghui Zhu, J. Wang, Jia-Jian Jiang, Alexander Hanbo Li, Rishav Chakravarti, Donald Weidner, Bing Xiang, and Zhiguo Wang. 2022. Generation-focused table-based intermediate pre-training for free-form question answering. In *AAAI Conference on Artificial Intelligence*.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for AMR-to-text generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1616–1626, Melbourne, Australia. Association for Computational Linguistics.

Yixuan Su, Zaiqiao Meng, Simon Baker, and Nigel Collier. 2021. Few-shot table-to-text generation with prototype memory. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 910–917, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Fei Wang, Zhewei Xu, Pedro Szekely, and Muhao Chen. 2022. Robust (controlled) table-to-text generation with structure-aware equivariance learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5037–5048, Seattle, United States. Association for Computational Linguistics.

Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. 2021. Tuta: Tree-based transformers for generally structured table pre-training. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery; Data Mining*, KDD '21, page 1780–1790, New York, NY, USA. Association for Computing Machinery.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. *EMNLP*.

Jingfeng Yang, Aditya Gupta, Shyam Upadhyay, Luheng He, Rahul Goel, and Shachi Paul. 2022. TableFormer: Robust transformer modeling for table-text encoding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 528–537, Dublin, Ireland. Association for Computational Linguistics.

Pengcheng Yin, Wen-Ding Li, Kefan Xiao, A. Eashaan Rao, Yeming Wen, Kensen Shi, Joshua Howland, Paige Bailey, Michele Catasta, Henryk Michalewski, Alex Polozov, and Charles Sutton. 2022. Natural language to code generation in interactive data science notebooks. *ArXiv*, abs/2212.09248.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426, Online. Association for Computational Linguistics.

Yan Zhang, Zhijiang Guo, Zhiyang Teng, Wei Lu, Shay B. Cohen, Zuozhu Liu, and Lidong Bing. 2020. Lightweight, dynamic graph convolutional networks for AMR-to-text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2162–2172, Online. Association for Computational Linguistics.

A Data Statistics

We summarize the input type and number of examples in each dataset.

Dataset	Input	# Examples		
		Train	Validation	Test
ToTTo	Table	120,761	7,700	7,700
DART	KG	30,526	2,768	6,959
LogicNLG	Table	28,450	4,260	4,305
WebNLG	KG	18,102	872	1,862
E2E clean	MR	33,525	1,484	1,847

Table 7: Data statistics.

B Column-wise Linearization of (Sub)Table

A column-wise linearization of Table 1 is:

```
1  <title> Alma Jodorowsky </title>
2  <sub\_title> Filmography </sub\_title>
3  <table>
4    <column>
5      <col_header> Year </col_header>
6        <cell> 2014 </cell>
7        <cell> 2016 </cell>
8        <cell> 2017 </cell>
9    </column>
10   ... other columns ...
11 </table>
```

C Other Linearizations Used in Previous Papers

Table highlights : Our unified representation is motivated by ToTTo linearization, and hence they are very similar. The only difference is ToTTo uses `<page_title>` instead of `<title>` and `<section_title>` instead of `<sub_title>`.

KG triples : Given a set of triples {(William Wasmund, FIELD_GOALS, 0), (William Wasmund, EXTRA_POINTS, 0)}, an alternative linearization used in UnifiedSKG (Xie et al., 2022) is `William Wasmund : field goals : 0 | William Wasmund : extra points : 0`

Entire table : The alternative linearization used in LogicNLG (Chen et al., 2020a) for Table 1 is: `Given the table title of Alma Jodorowsky, Filmograph. In row 1 , the Year is 2014 , the Title is La ..., the Role is Solana, the Notes is TV ... In row 2 , ...`

Mearning representation : The alternative linearization we use for the example in Figure 1 is simply concatenating all the MRs: `name[Cocum], eatType[coffee shop], food[Italian], priceRange[cheap], familyFriendly[yes].`

D Implementation Details

In the zero- and few-shot experiments, we employ the T5-base model as the base model and train it for 30 epochs for both the source and target tasks. For the source task, we use a learning rate of 5e-5 and a batch size of 32, and for the target task, we use a learning rate of 2e-5 and a batch size of 8.

E More Multi-Task Results

We present more detailed multi-task results on each of the dataset in this section. The results are summarized in Table 8, 9, 10 and 11.

F More Few-shot Results

We present other few-shot results using more metrics in Figure 3, 4 and 5.

G Human Evaluation

We conducted a human evaluation on the few-shot ToTTo to WebNLG transferring experiment. Specifically, we randomly selected 50 WebNLG test data from the unseen schema and compared the performance of the 8-shot *src to tgt, unified* and *src to tgt, varied* models.

For each of the 50 samples, we generated texts using both models and asked three annotators to choose the better option based on factuality, coverage of the triples, and fluency. We received only two annotations for two of the samples as one of the annotators did not respond. For the remaining 48 samples, all three annotators reached a consensus on 21 of them (43.75%). Out of these 21 samples, the "unified" model received unanimous preference from the annotators in 15 cases (71.43%). If we consider the majority vote among the three annotators, then 75% of the results favored the "unified" model. The Fleiss Kappa value, which measures agreement among the three annotators, is around 0.23 (fair agreement).

H More Qualitative Study

We present additional few-shot predictions for models transferred from ToTTo to WebNLG and LogicNLG in Tables 12 and 13, respectively. We also provide error analysis under each example.

Model	Task	Linearization	METEOR	ROUGE-L	CIDEr	NIST	BLEU
CONTROL PREFIX (large)	MT	Alt	39.2	-	-	-	44.2
T5-base	ST	Unified	38.3	56.6	1.90	6.20	41.8
	ST	Alt	38.3	56.4	1.91	6.23	42.1
	MT	Unified	**38.6**	**57.0**	**1.94**	**6.31**	**42.9**
	MT	Varied	38.3	56.6	1.89	6.20	41.7
T5-3B	ST	Unified	38.5	56.7	1.92	6.30	42.8
	ST	Alt	38.5	56.5	1.94	6.31	42.5
	MT	Unified	**38.7**	**57.4**	**1.99**	**6.34**	**43.2**
	MT	Varied	38.3	56.8	1.89	6.21	41.8

Table 8: Test set performance on E2E clean.

Model	Task	Linearization	Overall		Overlap		Non-overlap	
			BLEU	PARENT	BLEU	PARENT	BLEU	PARENT
LATTICE (T5-base)	ST	Table-specific	48.6	-	56.6	-	40.8	-
UnifiedSKG (base)	ST	Alt	48.3	-	-	-	-	-
UnifiedSKG (base)	MT	Varied	45.3	-	-	-	-	-
UnifiedSKG (3B)	ST	Alt	49.0	-	-	-	-	-
Text2Text (3B)	ST	Alt	48.4	57.8	-	-	40.4	53.3
T5-base	ST	Unified	49.3	58.9	57.1	62.7	**41.9**	**55.3**
	MT	Unified	**49.7**	**59.2**	**57.7**	**63.2**	41.9	55.2
	MT	Varied	48.5	58.7	56.2	62.5	41.1	55.0
T5-3B	ST	Unified	49.4	58.9	57.1	62.7	42.0	55.3
	MT	Unified	**50.8**	**60.4**	**58.5**	**64.4**	**43.4**	**56.5**
	MT	Varied	50.2	59.5	57.5	63.2	43.2	55.9

Table 9: Development set performance on ToTTo.

Model	Task	Linearization	DART			WebNLG		
			BLEU ($\uparrow$)	METERO ($\uparrow$)	TER ($\downarrow$)	Seen	Unseen	All
UnifiedSKG (base)	ST	Alt	46.2	-	-	-	-	-
UnifiedSKG (3B)	ST	Alt	46.7	-	-	-	-	-
CONTROL PREFIX (large)	MT	Alt	52.0	0.41	0.43	67.0	55.6	61.8
T5-base	ST	Unified	48.6	**0.40**	0.45	65.4	50.1	**58.5**
	ST	Alt	49.0	**0.40**	0.45	**65.9**	49.5	58.2
	MT	Unified	**49.8**	**0.40**	0.44	64.9	**50.3**	58.3
	MT	Varied	48.1	0.39	0.45	64.1	50.2	57.9
T5-3B	ST	Unified	49.6	0.40	0.45	65.1	52.7	59.5
	ST	Alt	49.3	0.40	0.45	65.3	**53.5**	**60.0**
	MT	Unified	**50.2**	0.40	0.44	65.4	53.4	**60.0**
	MT	Varied	49.8	0.40	0.44	65.3	51.9	59.4

Table 10: Test set performance on DART and WebNLG.

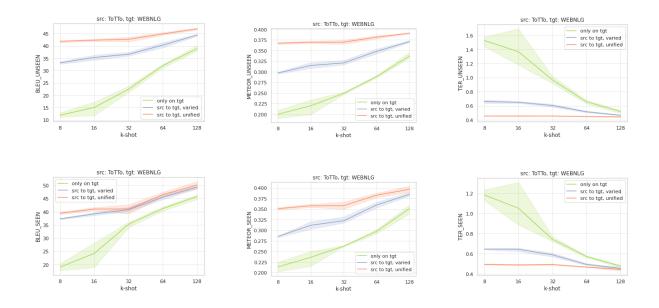| Model	Task	Linearization	Orientation	Surface-Level Fidelity			Logical Fidelity	
				BLEU-1	BLEU-2	BLEU-3	NLI-acc	SP-acc
GPT-TabGen	ST	Alt	row	48.8	27.1	12.6	68.7	42.1
DCVED	ST	Alt	row	49.5	28.6	15.3	76.9	43.9
T5-base	ST	Unified	column	52.8	34.9	24.3	79.6	45.2
	ST	Unified	row	53.3	35.4	24.7	84.7	45.8
	ST	Alt	row	**54.6**	**36.1**	**25.2**	**85.5**	45.9
	MT	Unified	column	53.8	35.8	25.1	78.7	**47.2**
	MT	Unified	row	54.4	**36.1**	**25.2**	80.4	46.3
	MT	Varied	row	53.9	35.5	24.7	84.2	46.3
T5-3B	ST	Unified	column	54.9	36.4	25.4	**88.4**	**49.8**
	ST	Unified	row	54.1	35.9	25.1	87.1	47.9
	ST	Alt	row	54.4	36.1	25.3	81.1	47.3
	MT	Unified	column	54.8	36.3	**25.4**	87.0	49.4
	MT	Unified	row	**55.1**	**36.4**	**25.4**	82.9	49.1
	MT	Varied	row	54.4	36.0	25.3	80.7	47.4

Table 11: Test set performance on LogicNLG.

Figure 3: Few-shot experiments of format transferring from ToTTo (table) to WebNLG (KG triples).

(a) BLEU

(b) METEOR

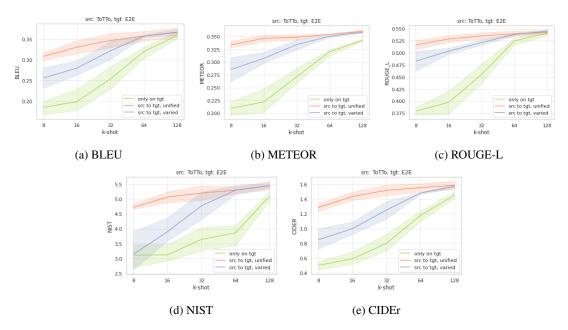(c) ROUGE-L

(d) NIST

(e) CIDEr

Figure 4: Few-shot experiment results of task transferring from ToTTo (table) to E2E (meaning representation).
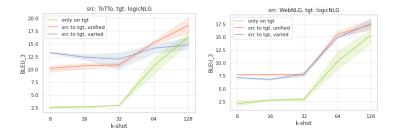

Figure 5: Few-shot experiment results of task transferring from ToTTo (table highlights description) or WebNLG (KG triples) to LogicNLG (logical inference on table).

k-shot =	Src to tgt, unified	Src to tgt, varied
	ToTTo (table) ⟶ WebNLG (KG) example	
8	Uruguay is led by Ral Fernando Sendic Rodrguez, who died in Montevideo, and by Daniel Martnez, a Spanish politician.	Ral Fernando Sendic Rodrguez died in Montevideo.
128	The leader of Uruguay is Ral Fernando Sendic Rodrguez who died in Montevideo, where Alfredo Zitarrosa died. The leader is Daniel Martnez who speaks Spanish.	Ral Fernando Sendic Rodrguez was the leader of Uruguay and Alfredo Zitarrosa died in Montevideo. Daniel Martnez was a politician who led the country in Spanish.
Groundtruth	Alfredo Zitarrosa died in Montevideo, Uruguay. Daniel Martinez is a political leader in Montevideo, and Raul Fernando Sendic Rodriguez is a leader in Uruguay, where Spanish is spoken.	
KG triples	`(Uruguay : leader name : Ral Fernando Sendic Rodrguez	Alfredo Zitarrosa : death place : Montevideo	Montevideo : country : Uruguay	Montevideo : leader name : Daniel Martnez (politician)	Uruguay : language : Spanish language`	
Error analysis	When sample size is 8, the "unified" model generation contains almost all nodes except Alfredo Zitarrosa, but the "varied" model output only contains one triple.	
8	Twilight (band) is a black metal band with Aaron Turner, and Old Man Gloom is a death metal band with electric guitar.	Twilight is a black metal music fusion genre.
128	Twilight (band) is associated with black metal, and Old Man Gloom is associated with death metal, where Aaron Turner played electric guitar.	Twilight is a genre of black metal music and Aaron Turner plays the electric guitar in Old Man Gloom. Death metal is a genre of black metal music.
Groundtruth	Aaron Turner is an electric guitar player who has played with the black metal band Twilight and with Old Man Gloom. Death metal is a musical fusion of black metal.	
KG triples	`(Twilight (band) : genre : Black metal	Aaron Turner : associated band/associated musical artist : Twilight (band)	Aaron Turner : associated band/associated musical artist : Old Man Gloom	Aaron Turner : instrument : Electric guitar	Black metal : music fusion genre : Death metal`	
Error analysis	In "unified" model output, the Twilight is always recognized correctly as a band, but in varied model output, it is wrongly predicted as a genre.	
8	Al Anderson (NRBQ band) played guitar in rock music, country music and blues music, while Bhangra (music) is a genre of music fusion.	Al Anderson is a guitarist with the NRBQ band.
128	Bhangra is a genre of music fusion, where Al Anderson plays guitar, Country music and Blues.	Al Anderson is an Indian country musician who plays guitar and plays Bhangra (music). Country music is a music fusion genre which is influenced by blues and rock music.
Groundtruth	Al Anderson plays the guitar for rock band NRBQ. Rock comes from country music which originates from blues. Bhangra music is a musical fusion of rock music.	
KG triples	`(Rock music : music fusion genre : Bhangra (music)	Al Anderson (NRBQ band) : instrument : Guitar	Al Anderson (NRBQ band) : genre : Rock music	Rock music : stylistic origin : Country music	Country music : stylistic origin : Blues`	
Error analysis	This is an negative example for "unified" model. When sample size is 8, the unified model generation covers more nodes compared to "varied" model generation. However, when sample size is 128, the "unified" model wrongly generates Anderson to play guitar, country music and Blue. This is a wrong fact. But "varied" model generation is able to be more aligned with the input triples.	

Table 12: Case study: few-shot prediction examples on WebNLG.

k-shot =	Src to tgt, unified	Src to tgt, varied
	ToTTo (highlighted cells) ⟶ LogicNLG (table) example	
8	Eric Turner, Ed King, James Jones, Pio Sagapolutele, Michael Jackson, Raymond Irvin, Shawn Wiggins, Brian Greenfield, Todd Jones, Todd Jones, and Elijah Austin played in the 1991 cleveland browns season.	In the 1991 cleveland browns season, Eric Turner, Ed King, Pio Sagapolutele and Raymond Irvin were the Defensive Tackle.
128	Raymond Irvin was the Defensive Back in the 1991 Cleveland Browns season.	Brian Greenfield, Todd Jones and Elijah Austin were the Defensive Tackle in the 1991 Cleveland Browns season.
8	Eric Turner, Ed King, James Jones, Pio Sagapolutele, Michael Jackson, Raymond Irvin, Shawn Wiggins, Brian Greenfield, Todd Jones, Todd Jones, and Elijah Austin played in the 1991 cleveland browns season.	In the 1991 cleveland browns season, Eric Turner, Ed King, Pio Sagapolutele and Raymond Irvin were the Defensive Tackle.
128	Raymond Irvin was the Defensive Back in the 1991 Cleveland Browns season.	Brian Greenfield, Todd Jones and Elijah Austin were the Defensive Tackle in the 1991 Cleveland Browns season.
8	Eric Turner, Ed King, James Jones, Pio Sagapolutele, Michael Jackson, Raymond Irvin, Shawn Wiggins, Brian Greenfield, Todd Jones, Todd Jones, and Elijah Austin played in the 1991 cleveland browns season.	In the 1991 cleveland browns season, Eric Turner, Ed King, Pio Sagapolutele and Raymond Irvin were the Defensive Tackle.
128	Raymond Irvin was the Defensive Back in the 1991 Cleveland Browns season.	Brian Greenfield, Todd Jones and Elijah Austin were the Defensive Tackle in the 1991 Cleveland Browns season.
Groundtruths	"Raymond Irvin is the second Defensive Back to get drafted", "Frank Conover is the third Defensive Tackle to get drafted", "Elijah Austin is the last Defensive Tackle to get drafted", "Frank Conover has an Overall that is 56 higher than Michael Jackson", "Shawn Wiggins is the second Wide Receiver to get drafted"	

player	Eric Turner	Ed King	James Jones	Pio Sagapolutele	Michael Jackson	Frank Conover	Raymond Irvin	Shawn Wiggins	Brian Greenfield	Todd Jones	Elijah Austin
position	Defensive Back	Guard	Defensive Tackle	Defensive Tackle	Wide Receiver	Defensive Tackle	Defensive Back	Wide Receiver	Punter	Offensive Tackle	Defensive Tackle

Input table (above)

Error analysis	Similar to our analysis in Section 4.3, the "unified" model generation is more like description when sample size is 8. Again this is because the source task is ToTTo, which is a task to generate surface-level description of table contents. The "unified" model transfers this learned knowledge better, and hence generates sentences that are more like descriptions. When sample size is 128, both models generate similar contents.

Table 13: Case study: few-shot prediction examples on LogicNLG.